

Desenvolvimento de uma Ferramenta para Detecção e Classificação de Padrões em Sequências de Imagens de Fenologia

Ricardo Elizeu Neto e Jurandy Almeida

Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo – UNIFESP

12247-014, São José dos Campos, SP – Brazil

Email: {ricardo.elizeu, jurandy.almeida}@unifesp.br

Resumo—Fenologia, ciência que relaciona ciclos biológicos com o clima, tem ganhado grande importância como o indicador mais simples e confiável dos efeitos das mudanças climáticas sobre plantas e animais. Uma das iniciativas de destaque relacionadas à área, o projeto e-Phenology, é uma proposta multidisciplinar que visa incorporar resultados de pesquisas correntes sobre fenologia tropical no Brasil com novas tecnologias provenientes de pesquisas em computação. Neste trabalho, foi desenvolvida uma ferramenta para auxiliar peritos da área de fenologia a detectar e classificar padrões em sequências de imagens de vegetação coletadas por câmeras digitais.

Abstract—Phenology, a science that relates biological cycles to climate, has gained importance as the simplest and most reliable indicator of the effects of climate change on plants and animals. An important initiative in this area, the e-Phenology project, is a multidisciplinary proposal that aims to incorporate results from current research on tropical phenology in Brazil with new technologies from computer science research. In this work, it was developed a tool for supporting experts in the phenology area to detect and classify patterns in vegetation image sequences collected by digital cameras.

I. INTRODUÇÃO

Fenologia é o estudo de fenômenos naturais recorrentes e sua relação com o clima [1]. Nos últimos anos, tal área tem ganho importância como o indicador mais simples e confiável dos efeitos das mudanças climáticas sobre plantas e animais [2], [3].

O e-Phenology¹, projeto voltado ao monitoramento de fenologia, é uma proposta multidisciplinar que combina pesquisas em computação e fenologia, visando incorporar resultados recentes de pesquisas correntes sobre fenologia tropical no Brasil [4] com novas tecnologias para detectar mudanças ambientais, e novas técnicas de observação remota da fenologia.

A observação direta dos ciclos das plantas impõe uma série de dificuldades aos observadores, especialmente nos ambientes tropicais, onde não há mudança abrupta ou marcante entre estações. Além disso, a observação em campo é geralmente conduzida por mais de um observador, o que leva a um viés em relação à percepção da mudança das plantas [5], [6].

Para aumentar a gama de locais e espécies de estudo, e a precisão das observações fenológicas, câmeras digitais têm

sido aplicadas com sucesso. Tais sensores de imagem de múltiplos canais fornecem medidas para estimar mudanças em eventos fenológicos, como brotamento e senescência [7].

Nesse contexto, a computação é uma grande aliada dos peritos em fenologia, fornecendo-lhes ferramentas para auxiliar nessa tarefa [8]–[10]. Por exemplo, algoritmos de processamento de imagens e aprendizado de máquina podem ser usados para detectar e classificar padrões em sequências de imagens da vegetação e identificar regiões que correspondem a espécies similares ou grupos funcionais [11]–[13].

Esse processo requer o uso de descritores. Um descritor pode ser visto como um par, composto por (1) um algoritmo capaz de codificar padrões fenológicos de regiões das imagens associadas a indivíduos de espécies de plantas em vetores de características; e (2) uma medida de similaridade para comparar regiões das imagens associadas a indivíduos de espécies similares ou grupos funcionais a partir de seus vetores de características [14].

O sucesso dessa tarefa está diretamente ligado aos descritores utilizados. Por um lado, várias pesquisas vêm sendo realizadas visando a especificação de descritores cada vez mais eficazes. Por outro lado, a maioria dos descritores propostos não estão integrados em uma ferramenta amigável, dificultando a sua utilização por peritos em fenologia. Por isso, existe uma demanda por novas ferramentas que integrem diferentes descritores, tornando essas soluções mais práticas e aplicáveis.

Este trabalho visa preencher essa lacuna a partir do desenvolvimento de uma ferramenta que, por meio do uso de descritores, apoie os peritos da área de fenologia a detectar e classificar padrões em sequências de imagens da vegetação.

As principais contribuições deste trabalho são (1) disponibilizar uma ferramenta de auxílio à peritos em fenologia, na detecção e identificação de padrões em sequências de imagens da vegetação, com o objetivo de reduzir as dificuldades relacionadas à percepção de mudanças das plantas; e (2) aperfeiçoar o processamento e a análise de sequências de imagens de fenologia geridas pelo projeto e-Phenology.

II. CONCEITOS BÁSICOS

A. Descritores de Imagem

Para a detecção e classificação de padrões em sequências de imagens, é necessário o uso de descritores. Assim, para

¹<http://www.recod.ic.unicamp.br/e-phenology/> (último acesso em 04 de setembro de 2017)

que o funcionamento da ferramenta possa ser compreendido, uma análise da definição e do funcionamento de descritores é necessária. Um descritor pode ser visto como [14]:

- 1) Um algoritmo capaz de extrair um vetor de características de uma imagem, a partir das suas propriedades visuais. Tais propriedades são diferentes para cada tipo de descritor (cor, textura ou forma).
- 2) Uma medida de similaridade entre duas imagens, a partir de seus vetores de características. Pode ser definida como uma função de distância entre esses vetores das imagens em questão.

A Figura 1 ilustra o uso de um descritor D para calcular a similaridade entre duas imagens \hat{I}_A e \hat{I}_B . Primeiro, o algoritmo ϵ_D é usado para extrair os vetores de características $\vec{v}_{\hat{I}_A}$ e $\vec{v}_{\hat{I}_B}$ associados às imagens. Depois, a função de distância δ_D é usada para determinar o valor da similaridade s entre as imagens a partir de seus vetores de características.

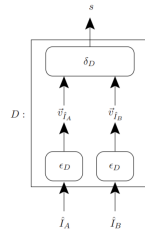


Figura 1. Uso de um descritor para comparar duas imagens [14].

Para o desenvolvimento da ferramenta, foram utilizados descritores de cor e de textura. Esses descritores foram escolhidos por se mostrarem os mais adequados ao monitoramento de fenologia a partir de sequências de imagens [8], [9].

B. Sistema e-Phenology

O projeto e-Phenology consiste de proposta multidisciplinar, que relaciona pesquisas da área de fenologia com as novas tecnologias provenientes de pesquisas em computação. Tal projeto foi iniciado em 2011 e já conta com diversas funcionalidades que auxiliam peritos em fenologia a monitorar e analisar dados coletados da vegetação.

Uma das funcionalidades desse sistema é a extração de séries temporais de sequências de imagens de uma região. Uma série temporal pode ser definida como um vídeo da vegetação, uma vez que ela é formada por imagens captadas em um intervalo fixo de tempo, por câmeras digitais.

Tais séries temporais podem ser exportadas para arquivos, e também podem ser utilizadas para gerar uma imagem em ritmo visual [8]. Ritmo visual é uma técnica que codifica a variação temporal da mudança da cor de *pixels* amostrados ao longo de uma linha específica [15], como mostra a Figura 2. Tal técnica pode ser utilizada para a detecção e classificação de padrões em uma sequência de imagens da vegetação. A sequência de imagens, tratada como uma série temporal, pode ser convertida em uma imagem em ritmo visual, que por sua vez pode ser analisada por descritores de imagem que extraíam seu vetor de características.

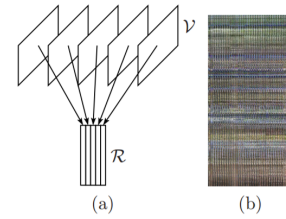


Figura 2. Ritmo visual: (a) simplificação de um conteúdo de vídeo através do mapeamento cada quadro para uma coluna de uma imagem; (b) um exemplo real produzido pela amostragem da linha central vertical da sequência de imagens da vegetação [8].

Portanto, o sistema e-Phenology e suas funcionalidades são tópicos de grande relevância ao trabalho, pois pretende-se incluir nele a ferramenta desenvolvida neste trabalho. Assim, o sistema será capaz de verificar a similaridade entre indivíduos da vegetação, para que os peritos possam ter um auxílio maior ao classificar tais indivíduos de acordo com a espécie.

III. IMPLEMENTAÇÃO

As classes necessárias para a ferramenta foram desenvolvidas em Java, pois tal linguagem permite uma maior reutilização e manutenção de códigos, por ser orientada a objetos. Primeiramente, foram definidas quais seriam os requisitos para a implementação, analisando as funções que a ferramenta deveria possuir. Assim, foi estabelecido que classes com as seguintes características deveriam ser implementadas:

- **Deteção de Regiões de Interesse.** Desenvolvimento de classes que possam realizar a detecção de quais regiões de uma imagem são relevantes. Assim, as classes recebem uma imagem, e realizam a aplicação de uma máscara. Desta forma, apenas as regiões mascaradas são utilizadas nas análises.
- **Extração de Características.** A implementação de classes que realizam a extração do vetor de características de uma imagem, ou seja, a implementação de algoritmos relacionados aos descritores de imagem.
- **Quantização de Cor.** Realizada em descritores de cor, a quantização geralmente recebe como parâmetro uma dimensão, e a partir dela quantiza uma imagem, comprimindo sua gama de cores. Também é levado em conta o sistema de cor desejado. Para isso, é necessária a implementação de uma classe com essa função.
- **Construção de Histogramas.** Como todos os descritores retornam como resultado um histograma, é necessário o desenvolvimento de uma classe que tem a função de manipulá-lo, realizando sua construção e normalização.
- **Adjacência de Pixels.** Alguns descritores, tanto de cor quanto de textura, realizam comparações e operações entre *pixels* adjacentes (vizinhos). Por esse motivo, é importante implementar classes que localizam todos os adjacentes de um *pixel*, de acordo com a distância e com o formato desejado para análise (circular ou retangular, por exemplo).

- **Conversão entre Sistemas de Cor.** A implementação de uma classe que tenha a função de converter de um sistema de cor para outro, visto que alguns descritores de cor realizam análises utilizando diferentes sistemas.

Assim, foi elaborado um diagrama de classes UML da ferramenta. Para uma melhor visualização, foram omitidos os atributos e métodos das classes. Essa versão simplificada do diagrama pode ser vista na Figura 3.

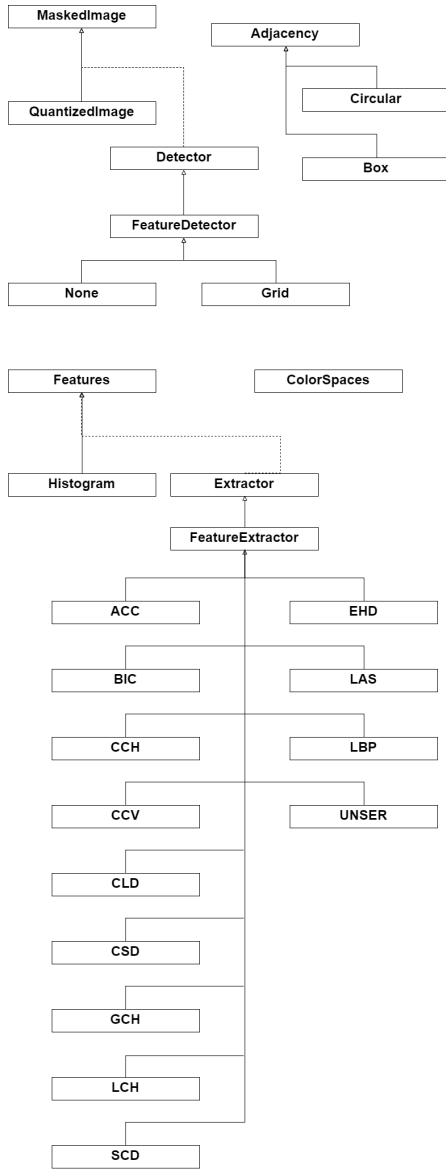


Figura 3. Diagrama de classes UML simplificado da ferramenta.

Com isso, foram desenvolvidas as classes que estendem as classes abstratas **FeatureDetector** e **FeatureExtractor**. Foram implementadas duas classes para detecção de regiões de interesse nas imagens, que estendem a **FeatureDetector**, que por sua vez implementa a interface **Detector**. São elas:

- **None.** Esta classe considera que toda a imagem é de interesse. Desta forma, a classe mascara todos os *pixels* da imagem.

- **Grid.** Esta classe divide a imagem em blocos menores, de acordo com um parâmetro de entrada. Como resultado, a classe retorna um **ArrayList** contendo tais blocos separadamente.

Como classes que estendem a **FeatureExtractor**, que implementa a interface **Extractor**, foram implementados os descritores de cor e de textura escolhidos.

Ao todo, foram implementadas 27 classes, como mostra a Figura 3, sendo:

- 1) 2 classes relacionadas à detecção de *pixels* de interesse em uma imagem (**None** e **Grid**);
- 2) 9 descritores de cor: ACC [16], BIC [17], CCH [17], CCV [18], CLD [19], CSD [19], GCH [20], LCH [20], SCD [19];
- 3) 4 descritores de textura: EHD [19], LAS [21], LBP [22], Unser [23];
- 4) 12 classes auxiliares.

Durante o desenvolvimento da ferramenta, procurou-se construir as classes buscando garantir a eficiência e qualidade do ponto de vista de engenharia de *software*. Como exemplo, pode-se analisar o Algoritmo 1, que contém a implementação do método principal do descritor de cor BIC (*Border-Interior Classification*). Todos os outros descritores seguiram o mesmo molde de implementação.

Algoritmo 1. Método de extração do vetor de características do descritor BIC

```

public ArrayList<Features<Integer>> extract(ArrayList<MaskedImage> detected) {
    int size = dim * dim * dim;
    ArrayList<Features<Integer>> features = new ArrayList<Features<Integer>>(0);
    int same_color;
    Adjacency adj = new Circular(1.0);
    for(MaskedImage original : detected){
        //Tratamento da imagem
        QuantizedImage quantized = new QuantizedImage(original, dim, ColorSpaces.TYPE_RGB);
        //Operacoes com os pixels
        ArrayList<Integer> border = new ArrayList<Integer>();
        ArrayList<Integer> interior = new ArrayList<Integer>();
        for(Map.Entry<Point, Integer> entry : quantized.getPixelsEntrySet()){
            Point pt = entry.getKey();
            int color = quantized.getCompressedColor(entry.getValue());
            same_color = 0;
            for(Point a : adj.getAdjPoints()){
                Point pt2 = new Point(pt.x + a.x, pt.y + a.y);
                int aux = quantized.getColor(pt2);
                int color2 = ((aux != -1) ? (quantized.getCompressedColor(aux)) : -1);
                if((color2 != -1) && (color == color2))
                    same_color ++;
            }
            if(same_color == 4)
                interior.add(color);
            else
                border.add(color);
        }
        //Construcao de um histograma
        Histogram[] histogram = new Histogram[2];
        histogram[0] = new Histogram(size, interior);
        histogram[1] = new Histogram(size, border);
        histogram[0].add(histogram[1]);
        histogram[0].normalize(quantized.maskSize());
        features.add(histogram[0]);
    }
    return features;
}
  
```

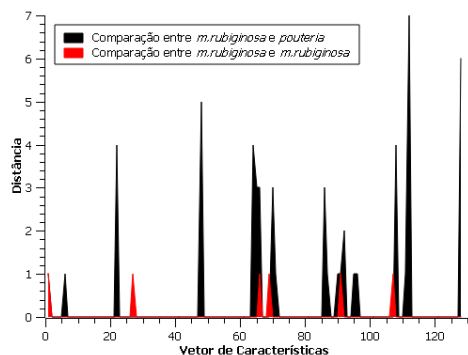


Figura 4. Gráfico contendo as distâncias entre vetores de características.

IV. RESULTADOS OBTIDOS

A fim de se atestar o funcionamento correto da ferramenta, foram realizados testes aplicando os descritores implementados em imagens em ritmo visual. Nele, é feita uma comparação entre dois indivíduos de mesma espécie e dois de espécies diferentes, a partir da distância Euclidiana entre seus vetores de características. A Figura 4 contém os resultados obtidos com o descritor BIC para as espécies *m.rubiginosa* e *pouteria*.

Analisando o gráfico da Figura 4, pode-se perceber que, ao comparar dois indivíduos da mesma espécie (em vermelho), a distância entre os elementos do seu vetor de características é igual a zero em praticamente todos os casos, salvo em determinados pontos onde o seu valor é relativamente baixo. Para duas espécies diferentes (em preto), mesmo com a maioria das comparações entre os elementos sendo iguais a zero, a distância atinge valores altos em alguns pontos, tendo como distância máxima o valor 7. Esse padrão de comportamento foi também observado nos demais descritores. Assim, pode-se dizer que a ferramenta opera como o esperado, podendo verificar se dois indivíduos são ou não da mesma espécie.

V. CONCLUSÃO

Neste trabalho, foram obtidos resultados adequados aos seus objetivos. Assim, a ferramenta desenvolvida pode aperfeiçoar a área de processamento de imagens do projeto e-Phenology e auxiliar peritos em fenologia em suas tarefas. A implementação foi feita buscando produzir códigos manuteníveis e modulares, garantindo a qualidade do ponto de vista de engenharia de *software*.

AGRADECIMENTOS

Os autores agradecem aos pesquisadores Dra. Leonor Patrícia Cerdeira Morellato, Dr. Ricardo da Silva Torres e Bruna de Costa Alberton pela assistência com os dados utilizados neste trabalho; e também às agências de fomento FAPESP (processo 2016/06441-7) e CNPq (processo 423228/2016-1) pelo apoio financeiro.

REFERÊNCIAS

[1] M. D. Schwartz, *Phenology: An Integrative Environmental Science*, 2nd ed. Springer, 2013.

[2] C. Parmesan and G. A. Yohe, "A globally coherent fingerprint to climate change impacts across natural systems," *Nature*, vol. 421, pp. 37–42, 2003.

[3] G. R. Walther, "Plants in a warmer world," *Perspectives in Plant Ecology Evolution and Systematics*, vol. 6, pp. 169–185, 2004.

[4] L. P. C. Morellato, M. G. G. Camargo, and E. Gressler, "A review of plant phenology in south and central america," in *Phenology: An Integrative Environmental Science*, M. D. Schwartz, Ed. Springer, 2013, chapter 6, pp. 91–113.

[5] L. P. C. Morellato, M. G. G. Camargo, F. F. D'Êça Neves, B. G. Luize, A. Mantovani, and I. L. Hudson, "The influence of sampling method, sample size, and frequency of observations on plant phenological patterns and interpretation in tropical forest trees," in *Phenological Research*, I. L. Hudson and M. R. Keatley, Eds. Springer, 2010, chapter 5, pp. 99–121.

[6] C. A. Polgar and R. B. Primack, "Leaf-out phenology of temperate woody plants: From trees to ecosystems," *New Phytologist*, vol. 191, pp. 926–941, 2011.

[7] B. Alberton, J. Almeida, R. Henneken, R. da S. Torres, A. Menzel, and L. P. C. Morellato, "Using phenological cameras to track the green up in a cerrado savanna and its on-the-ground validation," *Ecol. Inform.*, vol. 19, pp. 62–70, 2014.

[8] J. Almeida, J. A. Santos, B. Alberton, L. P. C. Morellato, and R. S. Torres, "Phenological visual rhythms: Compact representations for fine-grained plant species identification," *Pattern Recogn. Lett.*, vol. 81, pp. 90–100, 2016.

[9] F. A. Faria, J. Almeida, B. Alberton, L. P. C. Morellato, and R. S. Torres, "Fusion of time series representations for plant recognition in phenology studies," *Pattern Recogn. Lett.*, vol. 83, Part 2, pp. 205–214, 2016.

[10] J. Almeida, D. C. G. Pedronette, B. Alberton, L. P. C. Morellato, and R. S. Torres, "Unsupervised distance learning for plant species identification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 12, pp. 5325–5338, 2016.

[11] J. Almeida, J. A. dos Santos, B. Alberton, R. da S. Torres, and L. P. C. Morellato, "Applying machine learning based on multiscale classifiers to detect remote phenology patterns in cerrado savanna trees," *Ecol. Inform.*, vol. 23, pp. 49–61, 2014.

[12] J. Almeida, J. A. Santos, W. O. Miranda, B. Alberton, L. P. C. Morellato, and R. S. Torres, "Deriving vegetation indices for phenology analysis using genetic programming," *Ecol. Inform.*, vol. 26, pp. 61–69, 2015.

[13] F. A. Faria, J. Almeida, B. Alberton, L. P. C. Morellato, A. Rocha, and R. S. Torres, "Time series-based classifier fusion for fine-grained plant species recognition," *Pattern Recogn. Lett.*, vol. 81, pp. 101–109, 2016.

[14] R. da S. Torres and A. X. Falcão, "Content-based image retrieval: Theory and applications," *J. Theoretical and Applied Informatics*, vol. 13, no. 2, pp. 161–185, 2006.

[15] J.-S. Lee and T. Ebrahimi, "Perceptual video compression: A survey," *IEEE J. Sel. Topics in Signal Process.*, vol. 6, no. 6, pp. 684–697, 2012.

[16] J. Huang, R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR'97)*, 1997, pp. 762–768.

[17] R. O. Stehling, M. A. Nascimento, and A. X. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *ACM Intl. Conf. Information and Knowledge Management (CIKM'02)*, 2002, pp. 102–109.

[18] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *ACM Intl. Conf. Multimedia (ACM-MM'96)*, 1996, pp. 65–73.

[19] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 703–715, 2001.

[20] M. J. Swain and B. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[21] B. Tao and B. W. Dickinson, "Texture recognition and image retrieval using gradient indexing," *J. Visual Commun. Image Represent.*, vol. 11, no. 3, pp. 327–342, 2000.

[22] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recogn.*, vol. 29, no. 1, pp. 51–59, 1996.

[23] M. Unser, "Sum and difference histograms for texture classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 1, pp. 118–125, 1986.