



# UAV route planning for active disease classification

Kelen C. T. Vivaldini<sup>1</sup> · Thiago H. Martinelli<sup>2</sup> · Vitor C. Guizilini<sup>3</sup> · Jefferson R. Souza<sup>4</sup> · Matheus D. Oliveira<sup>2</sup> ·  
Fabio T. Ramos<sup>3</sup> · Denis F. Wolf<sup>2</sup>

Received: 12 January 2017 / Accepted: 10 July 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

## Abstract

Eucalyptus represents one of the main sources of raw material in Brazil, and each year substantial losses estimated at \$400 million occur due to diseases. The active monitoring of eucalyptus crops can help getting accurate information about contaminated areas, in order to improve response time. Unmanned aerial vehicles (UAVs) provide low-cost data acquisition and fast scanning of large areas, however the success of the data acquisition process depends on an efficient planning of the flight route, particularly due to traditionally small autonomy times. This paper proposes a single framework for efficient visual data acquisition using UAVs that combines perception, environment representation and route planning. A probabilistic model of the surveyed environment, containing diseased eucalyptus, soil and healthy trees, is incrementally built using images acquired by the vehicle, in combination with GPS and inertial information for positioning. This incomplete map is then used in the estimation of the next point to be explored according to a certain objective function, aiming to maximize the amount of information collected within a certain traveled distance. Experimental results show that the proposed approach compares favorably to other traditionally used route planning methods.

**Keywords** Route planning · UAV · Bayesian optimization · Rapidly-exploring random trees

✉ Kelen C. T. Vivaldini  
vivaldini@dc.ufscar.br

Thiago H. Martinelli  
thm@usp.br

Vitor C. Guizilini  
vitor.guizilini@sydney.edu.au

Jefferson R. Souza  
jrsouza@ufu.br

Matheus D. Oliveira  
matheusdellacroce@usp.br

Fabio T. Ramos  
fabio.ramos@sydney.edu.au

Denis F. Wolf  
denis@icmc.usp.br

<sup>1</sup> Department of Computer Science, Federal University of São Carlos (UFSCar), São Carlos, SP, Brazil

<sup>2</sup> Mobile Robotics Lab, Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP), São Carlos, SP, Brazil

<sup>3</sup> Australian Centre for Field Robotics, School of Information Technologies, University of Sydney, Sydney, Australia

<sup>4</sup> Department of Computer Science and Information Systems, Faculty of Computing (FACOM), Federal University of Uberlândia (UFU), Uberlândia, MG, Brazil

## 1 Introduction

According to the Brazilian Ministry of Agriculture, Livestock and Food Supply (MAPA 2015), eucalyptus and pine trees occupy 6,664,812 ha in Brazil, of which 76.6% correspond to eucalyptus and 23.4% correspond to pines. An estimated area of 2.9 million ha of eucalyptus forests provides a positive carbon footprint balance and also enables the exportation of \$2 billion per year. A 6.9 million ha increase is expected for 2020, which will promote an 8–10 million ton reduction in CO<sub>2</sub> emissions.

Among the problems commonly found in eucalyptus crops are the occurrence of diseases from the nursery stage until adulthood, in several locations and seasons. The harmful effects of diseases on eucalyptus are well-known and cause an estimated \$400 million loss per year, which totalizes \$2.8 billion in 7 years (Negro et al. 2014). *Ceratocystis fimbriata* *wilt* is one of the problematic diseases in eucalyptus crops due to its fast spreading. Once it has started, the fungal infestation is difficult to be controlled (Bedendo 1995). The wilt symptoms appear as a result of the blockage of the vessels by the mycelial growth of the fungus, which keeps water absorbed by the root system from adequately supplying the aerial part of the plant (Souza et al. 2015).

Because of continuous requests from producers and companies for the identification and control of such diseases, research has aimed at identifying behavioral changes and establishing appropriate strategies for disease control. Image processing, geoprocessing (GIS—Geographic Information System) and remote sensing have used remote images captured by satellites, piloted aircrafts and UAVs to evaluate and monitor crops. Remote sensing technology with UAVs provides spatial and temporal high-resolution, fast scanning of large areas and lower data acquisition costs in comparison to manned aircraft, satellites or airborne platforms, while achieving similar goals. Indeed, UAV surveys enable the use of remote images of very small pixel sizes, in the order of centimeters, thus improving image resolution in relation to other platforms (Jensen 2007; Candiago et al. 2015; Ponti et al. 2016).

UAVs are aerial vehicles that fly without a human pilot. They can be controlled remotely (by a certified pilot), partially autonomously (navigating according to a predefined route monitored by operators) or fully autonomously (containing various degrees of autonomy for carrying out certain tasks, such as path planning, detection and tracking of objects and decision-making) (Medeiro and Silva 2010; Bernardini et al. 2014; FAA 2016).

The Department of Defense (DoD) of the United States, the Federal Aviation Administration (FAA) and the European Aviation Safety Agency (EASA) have adopted the term unmanned aerial systems (UAS). A UAS comprises all the individual elements of a system, such as an UAV, a control station, data links and any other information necessary in a flight (Dalamagkidis et al. 2012), and is controlled by technicians or pilots in a control station. The data link transmits information between the UAV (send status) and the control station (send commands) (Park et al. 2017). Currently, the major focus of research into UAVs is on the increase of their autonomy (Grocholsky et al. 2006; Ludington et al. 2006; Medeiro and Silva 2010; Bernardini et al. 2014; Albore et al. 2015b), which consists in the transference of portions of the operator's decision-making process to the UAV itself.

Increased autonomy in the decision-making process can be obtained through navigation based on methods such as: computer vision; object detection, recognition and tracking during operation; autonomous refueling; communication and task-sharing with other UAVs. Therefore, a route planning is mandatory for the implementation of most newly required capabilities used to increase the autonomy of UAVs. Concomitantly, optimized routes significantly impact monitoring performance, and by extension all its subsequent applications. The flight route planning should enable the monitoring of surveyed areas as much as possible, given a series of constraints such as autonomy time, maximum altitude, distance to base and so forth.

This paper deals with the development of a novel route planning technique for active classification under uncertain conditions that uses Bayesian optimization (BO). The objective is to enhance the knowledge on visited areas and minimize the uncertainties about the classification of diseased trees while applying a restriction in the distance traveled. The Bayesian optimization approach combines environment representation, perception and route planning into a single framework. Opposed to previous techniques, that utilize greedy way-point solutions to acquire new observations, we consider information in a continuous sampling space, taking into account predictions based on data accumulated over time. At each new iteration, the next samples to be collected are selected according to an incomplete model composed of data acquired in previous iterations. The model is then improved via the incorporation of new samples, with no derivatives or knowledge about the underlying function. Therefore, the acquisition function must focus on the minimization of these unclassified areas to produce a more accurate map of the environment. The proposed approach improves on previous work (Vivaldini et al. 2016) taking into account that vehicles have time constraints due to energy for real applications. In this way, the distance constraint on the acquisition function that enforces a maximum distance traveled before the vehicle returns to the base becomes essential avoiding losses of the equipment and ensuring maximum use of the autonomy flight time. We also performed experiments with different sampling strategies and route planning algorithms, to determine the combination that provides the best results in an active classification scenario.

A probabilistic model is incrementally built on top of collected data, and the vehicle determines the next point to be explored by minimizing a certain objective function over the current incomplete model. The goal is to maximize the amount of information collected in a given traveled distance, determined by UAV autonomy, to ensure the mission is completed safely. The predictive mean and variance are used in a trade-off scenario between exploration–exploitation in a principled manner, according to the Bayesian method. The proposed methodology identifies and monitors any visually detectable pathogen, and the *Ceratocystis fimbriata* disease in real eucalyptus crops was chosen as a particular case study for experimental validation, however the same technique can be applied to a multitude of different tasks. Some examples include search and rescue (Liu 2016), surveillance (Witwicki et al. 2017), environment exploration (Tai et al. 2017) and forest fire detection (Ghamry et al. 2016). Similarly, the introduction of new sensors (i.e. multi-spectral and infrared cameras) can greatly increase the amount of information available for classification, thus producing better models and allowing the detection of a larger range of patterns.

## 2 Related work

Most techniques currently available in the literature tackle the problem of aerial surveillance and monitoring from two different perspectives: either based on artificial targets, that are defined prior to the beginning of navigation (i.e. no classification model); or focusing solely on the task of UAV image processing, with no concern for the trajectory traveled between sample points (i.e. no route planning). The framework proposed in this paper is hybrid, in the sense that it combines both classification and route planning to produce an active method of trajectory selection that is constantly adapting to the flow of new information.

In contrast to currently available hybrid methods, our approach does not require environment modification of any kind, since it learns a classification model purely based on sample images, and thus can be trained to search for any kind of pattern. Below we provide a brief historical overview of a few methods for aerial image classification and UAV route planning proposed over the years, highlighting their different advantages and drawbacks to show how our proposed approach can address some of these limitations.

### 2.1 Classification methods

Reid et al. (2011) developed an automated approach for vegetation classification in natural environments that is based on UAV images. Color and texture descriptors were extracted on a frame-by-frame basis for the construction of an appearance-based representation, classified by a novel multi-class generalization of the Gaussian process (GP). This new classification structure was trained and validated with manually labeled data and used in the construction of a probabilistic map of vegetation types. Tests were performed in an infested region of Northern Queensland—Australia, and results show accuracies of up to 88% amongst four tree classes with two invasive tree species.

Pérez-Ortiz et al. (2016) proposed a novel system for weed monitoring in sunflower crops using UAV images. The authors apply machine learning paradigms to minimize the intervention of the final user while studying the effects of three different parameters: flight altitude, choice of sensor and the use of previously trained models at different heights. Results show that it is possible to train a model at lower heights in a sub-portion of the experimental field and then apply it successfully to the rest of the field using a higher altitude flight.

### 2.2 Route planning methods

When route planning is considered, Lavalle and Kuffner (2000) proposed a method called rapidly-exploring random trees (RRT), which iteratively builds a graph between a

source point and a goal point, creating a path for robotic navigation. Expanding on that, Yang et al. (2013) developed an algorithm that combines RRT with occupancy maps generated by a Gaussian process. The method consists of a path planner that collects information about a searching area, focusing on portions with higher uncertainty and following an unorganized geometry.

Turker et al. (2015) used Simulated Annealing (SA) for the path planning of an UAV in a 2D scenario with random obstacles. Simulated results demonstrated that SA generates acceptable solutions, being capable of avoiding obstacles while following the overall planned trajectory. Ho and Liu (2010) developed an SA framework for robot path planning based on a Voronoi diagram (VD), Bezier curve (BC) and the Dijkstra algorithm (DA) to obtain the shortest smooth path. VD was applied to find a collision-free path, BC smoothed it and DA found the shortest path to be executed by the robot. An UAV route planning technique based on the genetic simulated annealing (GSA) algorithm was proposed in Meng and Xin (2010). A digital elevation map (DEM) produces a smooth flight surface, whereas the Genetic Simulated Annealing (GSA) algorithm plans the route of the UAV on this surface.

Weinstein and Schumacher (2007) formulated a Mixed Integer Linear Program (MILP) model for the task allocation problem based on a vehicle routing problem with time windows (VRPTW), adding various constraints to represent a variety of scenarios to the precise engagement of Intelligence, Surveillance and Reconnaissance (ISR). The focus is clustering targets, and implementing this information into the MILP to optimally assign UAVs to targets, considering a single depot and many target locations. They compare computation times and solutions for three different cost functions to be minimized: total distance of all routes, makespan and total time. Karakaya (2014) presented a modified Max–Min Ant System (MMAS) algorithm that calculates minimal distances covering a larger number of targets in a predefined flight range. The proposed method was compared with a Nearest Neighbour (NN) heuristic, and results show an increase of up to 10% in the number of covered targets.

Kim et al. (2017) presents an approach to the drone-aided delivery and pickup planning of medication products. The routes are predetermined and the delivery and pickup orders are known (targets). The Operational Planning (OP) model was proposed for optimal drone flight schedule for each center, and the cost–benefit analysis method was introduced as a decision-making criterion. A computational analysis was conducted to compare the performance of the problem using the Partition method and the Lagrangian Relaxation algorithm, which produced a better performance than the model without these components. Park et al. (2017) uses a MILP model and Sequential Tasks Allocation Heuristic (STAHE) in the mission planner to determine the essential service of UAV

task allocation. A customer selects their desired route, limited to 4 specific locations, and the routes and split jobs to each location are predetermined, with the UAV then calculating its trajectory. The MILP model offers an optimal solution only to small models due to a slow computational time, while heuristic approximations can determine non-optimal solutions much more efficiently.

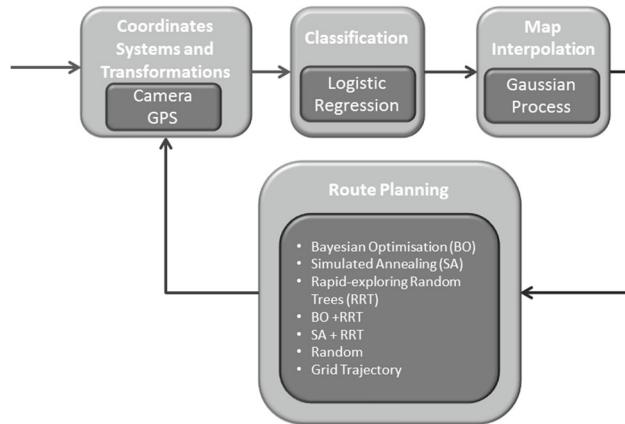
### 2.3 Hybrid methods

Albore et al. (2015b) presented a novel artificial intelligence planning-based approach for the autonomous production of an estimated map of pest abundance using UAVs. It verifies pests in a crop field and maps their spatial distribution using an online classifier, that allows decision-making during navigation. Markov random fields were integrated into the platform to update the information in the UAV's embedded sensors during runtime, with tests being conducted in the MORSE simulator. Albore et al. (2015a) expands on this work to propose an AI planning approach that integrates the planning process and calculates the probability distribution in a framework able to deal with task management and execution under time constraints. There is no prior knowledge of the environment state, and the policies are optimized and executed during the flight, showing an efficient trade-off between quality of selected sampling sites and planned navigation within the flight-time limits imposed by battery life. The platform integrates Markov Fields for knowledge representation, updated at runtime with information collected by the UAV.

Popović et al. (2017) proposed an Informative Path Planning (IPP) framework for active classification using UAVs for weed detection. They presented an adaptive strategy that generates dynamically feasible paths in continuous 3D space for information-theoretic objective, thus enabling the UAV to gather data efficiently. The proposed algorithm was validated in simulation against the lawnmower coverage and the sampling-based Rapidly exploring Information Gathering tree (RIG-tree), using the effects of different planning strategies. Results show that the proposed algorithm builds maps with over 50% lower entropy in the same amount of time when compared to the lawnmower coverage approach. Note that classification is performed using predetermined landmarks as targets, which greatly facilitates the classification process but is restricted to controlled environments.

## 3 Methodology

The framework adopted was first introduced in Vivaldini et al. (2016) and contains four modules (Fig. 1), namely: Coordinates Systems and Transformations; Classification; Map interpolation and Route Planning. Images obtained by the



**Fig. 1** Framework proposed contains four modules, namely: Coordinates Systems and Transformations; Classification; Map interpolation and Route Planning

UAV are first positioned in relation to a global coordinate system, whereas the Logistic Regression (LR) model classifies diseased trees from healthy trees and roads. A Gaussian process interpolates and creates a continuous map of the inspected area from the coordinates and values of classified points. Finally, the route planning module ensures a good coverage of the environment by selecting sample points that maximize information collected. We have modified the route planning module to add a constraint for traveled distance and included the RRT algorithm to generate the best trajectory to be executed between sampling points.

Three methods for route planning were adopted for experimental validation within the proposed framework, focusing on efficiency and performance. Tests were performed in the MORSE Simulator, which received real images from an UAV to emulate an eucalyptus crop scenario. As a result, the proposed framework can be equally used in both simulated and real environments, requiring only the adjustment of camera parameters and coordinate systems.

### 3.1 Coordinates systems and transformations

The UAV's orientation, intrinsic and extrinsic camera parameters, GPS coordinates and captured images were used as input information for the active route planning algorithm. Once the relationships have been obtained, each image is transformed into an equivalent normal view, which results in the generation of a new image and the mapping of each pixel location in relation to its GPS information. As a convention, the GPS information refers to the geodetic coordinate system, and the UAV rotation matrix follows Euler's XYZ convention.

### 3.2 Logistic regression

For the classification module, a LR (Hyttinen et al. 2015) model was trained to classify diseased trees from healthy trees and other structures. LR is quite robust to noise and avoids over-fitting through the regularization of  $L_2$  and  $L_1$  norms (Ng 2004). An advantage of such method is the possibility of obtaining the probability value of the predicted class, ranging from [0 to 1]. Therefore, confidence intervals can be set to each classification result as a way to model uncertainty in the final model. The prediction model can be easily updated by Stochastic Gradient Descent as new data becomes available.

#### 3.2.1 Overview

The LR model is a linear classifier that creates a model that maps relationships between a dependent variable  $Y$  and a series of independent variables  $X_1, X_2, \dots, X_n$  from training data. It uses only two possible states (0 or 1) for the dependent variable  $Y$ , as it depends on the occurrence of the event considered, and can be written as:

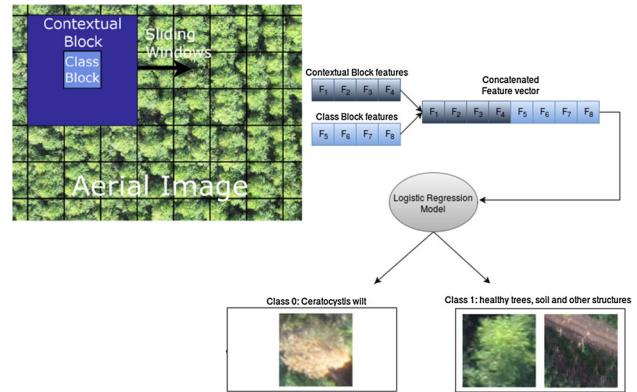
$$P(Y = 1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}} \quad (1)$$

The probability of an eucalyptus tree developing Ceratocystis wilt is represented by  $P$ , which is the conditional probability of  $Y$  when it assumes a value of 1. Parameters  $\beta_0, \beta_1, \beta_2, \dots, \beta_n$  are estimated using the Maximum Likelihood technique for the measurement of variations in probability proportions. Such coefficients combined with  $X$  are modeled by a sigmoidal curve, so that when  $\mu(x) \rightarrow +\infty$ , probability  $P(Y = 1)$ , and when  $\mu(x) \rightarrow -\infty$ ,  $P(Y = 0)$ .

#### 3.2.2 Contextual block classification

The Contextual Block Classification methodology (Souza et al. 2015) was used for the detection and classification of eucalyptus crops affected by the Ceratocystis wilt disease. A square sliding window (block) runs on the image and extracts visual features. Each block is surrounded by a larger contextual block (Fig. 2). Visual features are extracted from each block and its contextual block and concatenated to form the feature vectors.

In a remote sensing, for the extraction of the features vector, the spectrum of color and texture features are commonly applied to classification problems using machine learning (Reid et al. 2011). Both the RGB channels and the image converted to grayscale are used, and therefore each image's pixel can be represented by a 4-element vector



**Fig. 2** Contextual block methodology, adapted from Souza et al. (2015). A square sliding window (block) runs on the image and extracts visual features. Each block is surrounded by a larger contextual block. Visual features are extracted from each block and its contextual block and concatenated to form the feature vectors. And, finally, the LR classifier classifies trees from healthy trees and others structures

$p_{x,y} = \{red, green, blue, gray\}^T$ , where  $(x, y)$  are coordinate values of each index. Mean  $M = \frac{1}{|P|} \sum_{x,y \in P} e_{x,y}$  and variance  $V = \frac{1}{|P|} \sum_{x,y \in P} (e_{x,y} - M)^2$  values are employed for each sample pixel  $P$ . The mean and variance of the pixels for the image converted to CIELab color space were also used in the feature vector, since the brightness component  $L$  can be decoupled (Gonzalez and Woods 2002) and it captures variations in image brightness. A texture descriptor *Local Binary Patterns* (LBP) (Ojala 2002) is used in the 4-channel of vector  $p$ , in which each 4-connected neighbor generates a histogram of 16 bits that are inserted into the feature vector. Finally, entropy  $H(A)$ , which measures the amount of information on the grayscale image, is also inserted into the feature vector.

A manually classified map (ground-truth) assigns each feature vector to one of two classes, i.e., 1 for diseased trees or 0 for healthy trees and other structures. Finally, the LR classifier is trained based on this information to create a model that can probabilistically discern diseased trees in newly obtained images.

### 3.3 Gaussian process regression

A Gaussian process (Rasmussen and Williams 2006) is a Bayesian technique for non-parametric regression and classification. It is non-parametric because it does not maintain an explicit model of the underlying phenomenon, but learns the transformation between input and output directly from training samples (observations). It is Bayesian in the sense that it places *prior* distributions (hypotheses) on observed variables and updates these distributions to produce a *posterior* distribution as new data become available. It is a regression

technique because it produces continuous smooth outputs at arbitrary resolutions and indicates not only the best estimate given the current information, but also the uncertainty inherent to that particular estimate.

For the problem at hand, we assume a dataset  $\mathcal{D} = (X, \mathbf{y}) = (\mathbf{x}_i, y_i)_{i=1}^N$ , composed of  $N$  training inputs  $\mathbf{x}_i$  containing spatial coordinates for the observed environment and their respective probability of disease  $y_i$ , as given by a classification algorithm such as LR (Sect. 3.2). The correlation between the underlying function  $f(\cdot)$  and those probabilities is given by:

$$y(\mathbf{x}_i) = f(\mathbf{x}_i) + \epsilon, \quad (2)$$

where  $\epsilon$  is an independent noise component commonly modeled as a zero-mean Gaussian distribution with variance  $\sigma_n^2$ , i.e.  $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ . The other two functions, i.e., mean and covariance, are also used for the encoding of our prior knowledge in relation to the underlying function. The mean function  $m(\mathbf{x}; \boldsymbol{\theta}_m)$  represents the most probable value at each point in the input space and is commonly assumed to be a constant value  $\sigma_m$ . The covariance function  $k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}_k)$ , on the other hand, models the correlation between any two given points in the input space.

The literature reports a variety of well-established covariance functions (Rasmussen and Williams 2006) that rely on different techniques to model those correlations and are designed for specific modeling scenarios. Indeed, the selection of the best-suited covariance function for a particular problem is a field of research in and of itself, as shown in Duvenaud et al. (2013). We used the Rational Quadratic covariance function,<sup>1</sup> defined as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \left( 1 + \frac{(\mathbf{x}_i - \mathbf{x}_j)^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)}{2\alpha} \right)^{-\alpha} \quad (3)$$

where  $\boldsymbol{\theta}_k = (\Sigma, \alpha)$ , with  $\Sigma$  and  $\alpha$  being a length-scale diagonal matrix controlling smoothness. The parameters  $\boldsymbol{\theta} = (\sigma_m, \boldsymbol{\theta}_k, \sigma_n)$ , which serve as coefficients for the functions, are usually known as *hyperparameters* and are obtained by the maximization of the log-marginal likelihood function:

$$\log p(\mathbf{y}|X) = \log \mathcal{N}(\mathbf{y}|m(X; \boldsymbol{\theta}_m), K_{nn} + \sigma_n^2 I) \quad (4)$$

This equation provides a natural balance between data fit and model complexity through the Occam's Razor principle (Rasmussen and Williams 2006) for the avoidance of overfitting (i.e. excellent performance during training and poor

performance during testing). After the optimized hyperparameters are obtained, the value at a test point  $\mathbf{x}_*$  is given by a Gaussian distribution with mean and variance  $\mu_*$  and  $\mathcal{V}_*$ , respectively:

$$\mu_* = K_{*n} \left( K_{nn} + \sigma_n^2 I \right)^{-1} (\mathbf{y} - m(\mathbf{x})) \quad (5)$$

$$\mathcal{V}_* = K_{**} - K_{*n} \left( K_{nn} + \sigma_n^2 I \right)^{-1} K_{n*}, \quad (6)$$

where  $K_{nn}$  is the  $n \times n$  covariance matrix with  $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\theta}_k)$ ,  $K_{*n}$  is the covariance matrix between test and training points, and  $K_{**}$  is a diagonal matrix that encodes the covariance between test points. Such estimates are continuous, and inference can be made at an arbitrary resolution in a point of the input space. A common approach for the transformation of these estimates into valid probability distributions between two discrete classes (in our case, diseased trees or other structures) is the “squashing” of mean estimates by a response function, such as the *logistic function*, defined as:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \quad (7)$$

where  $x_0$  defines the sigmoid's x-value at its midpoint,  $L$  determines the curve's maximum value and  $k$  its steepness. Such parameters can be determined according to the training data and optimized alongside the GP hyperparameters. For the proposed framework, these parameters are determined during the training process based on ground-truth information, to determine the proper scale and sensitivity of the underlying problem. Once they are optimized, the online learning process takes place by introducing new points into an initially empty non-parametric model, obtained during navigation and representing observations of an unknown environment. Further training can then be conducted to improve the representativeness of the current model by refining its parameters, however we noticed during experiments that they are already stable enough, and further training did not improve results.

Note that, while the original Gaussian process implementation (Rasmussen and Williams 2006) has difficulties scaling up to larger datasets, since it has a computational complexity that increases cubically with the number of data points, over the years several extensions have been proposed to alleviate this limitation. In Snelson and Ghahramani (2006) an approximation is introduced, that uses a small subset of inducing points to project input data into a lower-dimensional manifold, thus decreasing complexity during training and inference. Stochastic variational inference is used in Hensman et al. (2013) to allow training using mini-batches of available data, so the entire dataset is never touched and online updates can be performed as new data is collected.

<sup>1</sup> In previous works (Souza et al. 2015), several different covariance functions were considered, and the Rational Quadratic produced better classification results.

These extensions would allow the GP framework to be used in much larger environments while still maintaining efficient routines for data incorporation and retrieval.

### 3.4 Bayesian optimization

#### 3.4.1 Overview

BO is a sequential design technique that searches for the maximum of an unknown function  $f(\cdot)$  that is too costly to be evaluated directly (Snoek et al. 2012), or too complex for analytical calculations. Instead, it employs the Bayes theorem to incrementally combine prior information with observations (samples) and produce new estimates of an underlying function  $f(\cdot)$  while attempting to reach its maximum. At each new iteration, the next samples to be collected are selected according to an incomplete model composed of data acquired in previous iterations. The model is then improved via the incorporation of new samples, with no derivatives or knowledge about the underlying function.

Here, we use a GP model as the prior function, with components that model both the mean value of  $f(\cdot)$  and its respective variance (uncertainty about the estimate). All observations are treated as noisy samples collected from an unknown function  $f(\cdot)$ . The next point to be sampled at each iteration is selected by the minimization of an intermediate and predetermined function  $h(\cdot)$ , henceforth referred to as the *acquisition function*. The selection of the acquisition function is crucial for a proper BO performance (Marchant and Ramos 2012), since it determines the intrinsic behavior that will lead to new samples at each iteration.

#### 3.4.2 BO for path planning

The GP framework (described in Sect. 2.3) contains results from the classification algorithm (LR) that range from 0, indicating diseased trees, to 1, indicating healthy trees or other structures. In-between values represent ambiguous areas, and 0.5 indicates complete uncertainty about the classification of a particular point. Therefore, the acquisition function must focus on the minimization of these unclassified areas to produce a more accurate map of the environment. We propose the codification of such behavior by the following acquisition function:

$$h(\mathbf{x}) = -\sigma_v^2 * \exp\left(-\frac{1}{2} \left(\frac{\mu - 0.5}{\sigma_l}\right)^2\right) \quad (8)$$

---

#### Algorithm 1 Continuous path Bayesian optimization

---

```

Require:  $f, h, \mathcal{C}$ 
for  $i = \{1, 2, 3, \dots, \text{iterations}\}$  do
    Find  $\beta^* = \arg \max_{\beta} r(\mathcal{C}(u, \beta)|h)$ 
     $\{\mathbf{x}, y\}_{\mathcal{C}} \leftarrow \mathcal{C}(u, \beta^*)|_{u=0}^1$ 
     $\mathcal{GP} \leftarrow \{\mathbf{x}, y\}_{\mathcal{C}}$ 
end for

```

---

where a Gaussian distribution with mean 0.5, amplitude  $\sigma_v$  and standard deviation  $\sigma_l$  is employed. Since we deal with minimization during the optimization process, the negative sign is used to flip the Gaussian distribution.

The traditional BO derivation is discrete, i.e. only the final destination of each iteration is taken into account. However, here we extend it to a continuous domain (Marchant and Ramos 2012), in which the trajectory between start and end points is also considered. This is of particular interest for the application at hand, because it enables the aircraft to obtain images as it navigates between points with no additional effort. A score  $s$  for each trajectory  $\mathcal{C}$  is calculated by the integration of the acquisition function over its length:

$$s(\mathcal{C}(u, \beta|h)) = \int_{\mathcal{C}(u, \beta)} h(u)du, \quad (9)$$

where  $\beta$  are trajectory coefficients and  $u = [0, 1]$ . If Eq. 9 cannot be analytically calculated,<sup>2</sup> approximations such as sampling or rectangle-rule quadrature (Stoer et al. 2002) may be used. Once the optimized trajectory  $\beta^*$  has been determined, samples  $\{\mathbf{x}, y\}_{\mathcal{C}}$  are obtained along the way (i.e. at fixed-length intervals) and added to the GP model as new training points. The process is then repeated for the production of a new optimized path based on the updated model (as shown in Algorithm 1).

### 3.5 Simulated annealing

Simulated Annealing (SA) is based on the physical process of metal cooling and the traditional optimization problem (Ingber and Rosen 1992). It uses a principle of evolving the solution over time, in which the annealing expression most used in the literature corresponds to liquid metals that are cooled to achieve a low-energy state. SA is a probabilistic algorithm that approximates the global optimum of a function. In other words, it sweeps all the search space to find a general solution. The cooling concedes small movements in the solution space, which eventually converges to a final result (Kirkpatrick et al. 1983). Algorithm 2 (Engelbrecht

---

<sup>2</sup> We employ line segments as the template for trajectory calculations, however Eq. 9 can be equally applied to any sort of curve, such as splines (Egerstedt and Martin 2001).

**Algorithm 2** Simulated annealing

---

**Require:**  $C$ ,  $NS$ ,  $T$  and **Output:**  $X^*$

```

while  $T > 0.1$  do
    while  $kNeighbors < NS$  do
        Generates a new solution  $X'$ 
        Calculates fitness  $f(X')$  and  $f(X)$ 
         $prob = 1/(1 + e^{\frac{f(X') - f(X)}{T}})$ 
        if  $(f(X') < f(X))$  or  $(prob > r)$  then
             $X^* \leftarrow X'$  and  $f(X) \leftarrow f(X')$ 
        end if
        if  $T > 2.5$  then
             $xNP \leftarrow XMAX$ 
             $yNP \leftarrow YMAX$ 
        end if
        if  $T < 1.5$  then
             $xNP \leftarrow (XMAX/2)$ 
             $yNP \leftarrow (YMAX/2)$ 
        end if
         $kNeighbors \leftarrow kNeighbors + 1$ 
    end while
     $T \leftarrow T * C$ 
end while

```

---

2006) shows the pseudo-code for the SA when applied in a path planning strategy.

### 3.5.1 SA for path planning

The SA algorithm is responsible for choosing the optimal goal points for an aircraft to visit during navigation, in order to minimize the distance travelled and ensure an adequate coverage of the inspected area.

For a new solution  $X'$ , SA will:

1. Assign value to neighbors in x-axes and y-axes, where the minimum and maximum values are selected empirically ( $xNeighbors$  and  $yNeighbors$  for global search and  $xNeighbors$  and  $yNeighbors$  for local search).
2. Generate a random number for  $xDistance = [-xNeighbors, xNeighbors]$  and  $yDistance = [-yNeighbors, yNeighbors]$ ,  $xDistance$  and  $yDistance$  are summed for the current solution.
3. Calculate the uncertainty average of the image in  $X$  and  $X'$ , e.g. determine the map values of  $X' - threshold$  until  $X' + threshold$  for the x-axis and the map values of  $X' - threshold$  until  $X' + threshold$  to the y-axis. 4) Calculate the average uncertainty value ( $aUncertainty$ ), which serves as input to the fitness function (Eq. 10) for the solution  $X'$ .

As reported in Sect. 3.4.2, the diseased trees are represented by a value of 0 and healthy trees or other structures are represented by a value of 1. Intermediate values are ambiguous or obscure areas, and 0.5 represents the lack of knowledge about the classification at a particular point. The SA fitness function aims to reduce this ambiguity and is useful for

the construction of a confidence map of classified regions. The fitness function codifies such behavior in the following way:

$$f(X) = |1 - |\exp(0.5 * (aUncertainty - 0.5)^2)|| \quad (10)$$

The above equation describes the objective function of the SA, where  $aUncertainty$  represents the uncertainty average of the image in position  $(x, y)$  on the current classification map. SA minimizes the objective function described in Eq. 10 for the choice of positions  $(x, y)$  with values close to 0.5.

In summary,  $X$  is the initial solution,  $X'$  is the new solution for each iteration of the algorithm (candidate),  $X^*$  is the best solution found and  $f$  is the objective function (fitness). The candidate solution  $X'$ , which suggests a position in a given neighborhood considering both coordinates  $x$  and  $y$  of the map, is then used as  $X$  in the next iteration.

### 3.6 Rapidly-exploring random trees

A Rapidly-exploring Random Tree (RRT) searches non-convex spaces with high dimensionality through the build of a space-filling tree to store possible paths. In comparison to other randomized algorithms, such as randomized potential fields and probabilistic roadmaps, RRT naturally extends to general non-holonomic planning problems (including kinodynamic representations (Donald et al. 1993)). An RRT iteratively expands through the application of control inputs that move the system towards randomly-selected points. Due to the use of Voronoi diagrams, RRT tends to explore unsearched areas (Donald et al. 1993).

The RRT algorithm is illustrated in Algorithm 3. In this algorithm,  $x_{initial}$  stands for a pre-existent tree to which vertices will be added.  $L$  is the initial number of vertices of the tree. This algorithm represents one iteration of the tree-building process. At each iteration, a new vertex is added to the tree, obeying the following procedure: First, a randomly generated vertex is created in the unsearched space. Then, a new edge is drafted between the new point and the nearest vertex in the tree. If the new edge does not cross any part of the searched space, it is added to  $x_{initial}$ .

**Algorithm 3** Creation of a random tree

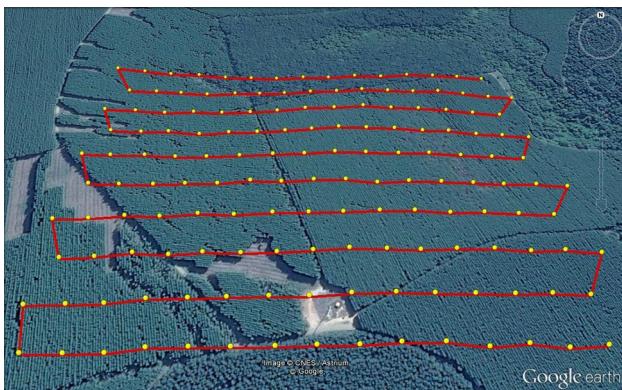
---

```

Create_Tree ( $x_{initial}$ ,  $L$ ,  $\Delta x$ )
for  $i = \{1, 2, 3, \dots, L\}$  do
     $x_{random} \leftarrow Random\_Position()$ 
     $x_{nearest} \leftarrow Nearest\_Vertex()$ 
     $x_{new} \leftarrow New\_Edge(x_{new}, \Delta x)$ 
     $Graph.include\_vertex(x_{new})$ 
     $Graph.include\_edge(x_{near}, x_{new})$ 
     $Return G$ 
end for

```

---



**Fig. 3** The UAV followed a predefined route on a farm containing eucalyptus trees to collect data. The final dataset consists of 154 images, from which we used 15 due to the presence of diseased trees for training and testing



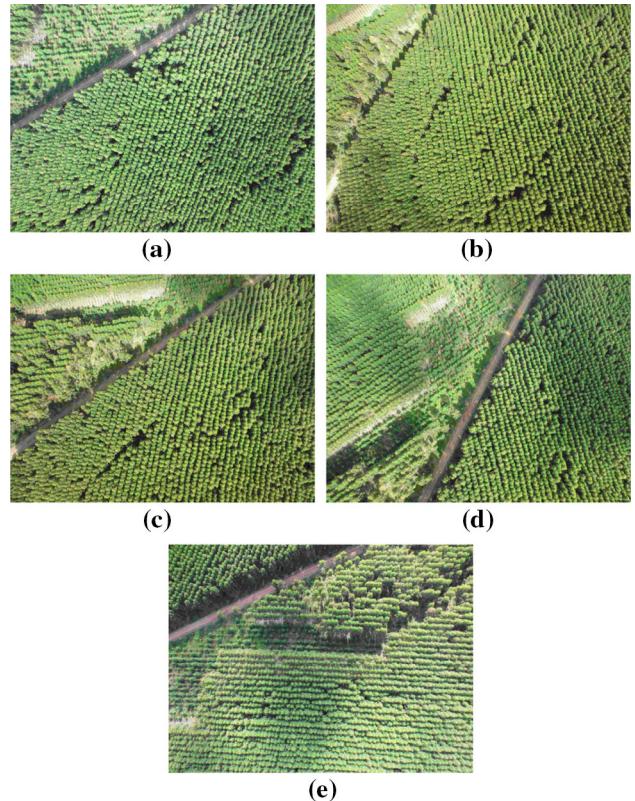
**Fig. 4** The eBee robotic platform used during experiments, to capture high-resolution images of eucalyptus crops

## 4 Platform simulation

We used the Robotic Operating System (ROS) and the Modular OpenRobots Simulation Engine (MORSE) as communication protocol and simulation platform, respectively. This configuration has been widely adopted for the testing and evaluation of robot software in several missions (Echeverria et al. 2011; Milliez 2014; Albore et al. 2015b; Degroote et al. 2016; Zhou et al. 2016; Park et al. 2017; Mullaonkar and Kumar 2014). ROS is a robotic meta-operating system that provides hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes and package management (Quigley et al. 2009). It has now become the standard communication protocol in robotics applications, due to its transparency and efficiency during data transfer between different software modules or machines.

MORSE is a fully open-source simulation suite based on Blender that aims at simplifying the definition and development of integrated complex robotics experiments. Blender simulates photo-realistic 3D worlds with an associated physics engine, bringing enough realism for the evaluation of complete sets of components within a wide range of application contexts (Echeverria et al. 2012). One of the advantages of using MORSE is that it applies the Software Architecture-In-the-Loop (SAIL) principle, where the same architecture used in the simulation can be directly applied to the real UAV, modifying only the simulated inputs of the data by the physical sensors and actuators (Lemaignan et al. 2014).

It is important to note that all images used in this work are from a real farm containing eucalyptus (Fig. 3), captured by an eBee robot (Fig. 4) with an IXUS 127 HS Canon - RGB camera at 890 m average altitude and a  $4608 \times 3456$  pixel resolution. Each image has an area of  $25,715 \text{ m}^2$  and represents different scenarios (Fig. 5). These images were then



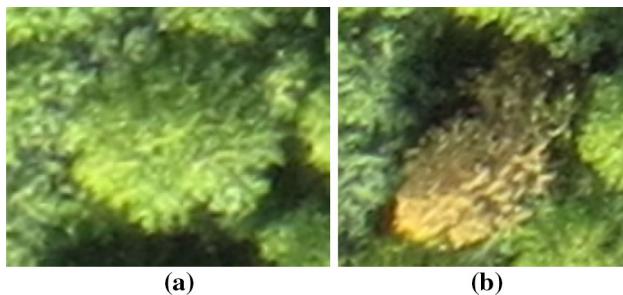
**Fig. 5** Examples of different scenarios from the test set were used as the texture in the simulated environment. The images are from a real farm. **a** Scenario1, **b** Scenario2, **c** Scenario3, **d** Scenario4, **e** Scenario5

used as textures in the simulated environment, so different route planning strategies can be efficiently tested multiple times under the same set of environmental circumstances.

For the testing environment, we adopted each scenario independently and added a simulated UAV (quadrotor) with Rotorcraft attitude, velocity and waypoint motion controller. The UAV was also equipped with a GPS, an IMU and a cam-



**Fig. 6** The UAV was added in the testing environment to capture images with pixel resolution  $4608 \times 3456$  at an average height of 890 m. The UAV was also equipped with a GPS and an IMU to provide information for the module Coordinates Systems and Transformations



**Fig. 7** Examples of: **a** healthy eucalyptus tree and **b** eucalyptus with Ceratocystis wilt

era, collecting images with pixel resolution  $4608 \times 3456$  at an average height of 890 m. Flight stabilization was obtained through actuator controls using a linear model available in the MORSE simulator, which allows it to perform the desired trajectory accurately.

The simulated UAV (Fig. 6) can then travel to these areas and obtain the necessary information from its sensors and camera, as a real vehicle would do in a real environment. Each captured image has an area of  $1200 \text{ m}^2$  and depicts a subset of the full-size image at arbitrary positions and altitude values.

## 5 Experimental results

Results were evaluated in all scenarios (Fig. 5) from the test set. The datasets provided sufficient examples of healthy eucalyptus crops and diseased trees for the evaluation of the experiments. As addressed in Sect. 1, eucalyptus trees have the biotic stress known as Ceratocystis Wilt (Fig. 7). The study related to diseases in eucalyptus trees was developed by Adimara B. Colturato and originally presented in Souza et al. (2015).

The LR model was trained and tested as the classification module (Sect. 5.1). The same classification module was included in the proposed framework for all tests, without

**Table 1** Hand-crafted features per channel extracted from the dataset

	R	G	B	Gray	RGB
Mean	1	1	1	1	0
Variance	1	1	1	1	0
CieLAB (mean)	0	0	0	0	3
CieLAB (variance)	0	0	0	0	3
LBP (histogram)	0	0	0	16	0
Entropy (mean)	0	0	0	0	1
Entropy (variance)	0	0	0	0	1
Total	2	2	2	18	8

modification. Different methods for the route planning module were applied to evaluate the performance of the proposed approach, as shown in Sect. 5.2).

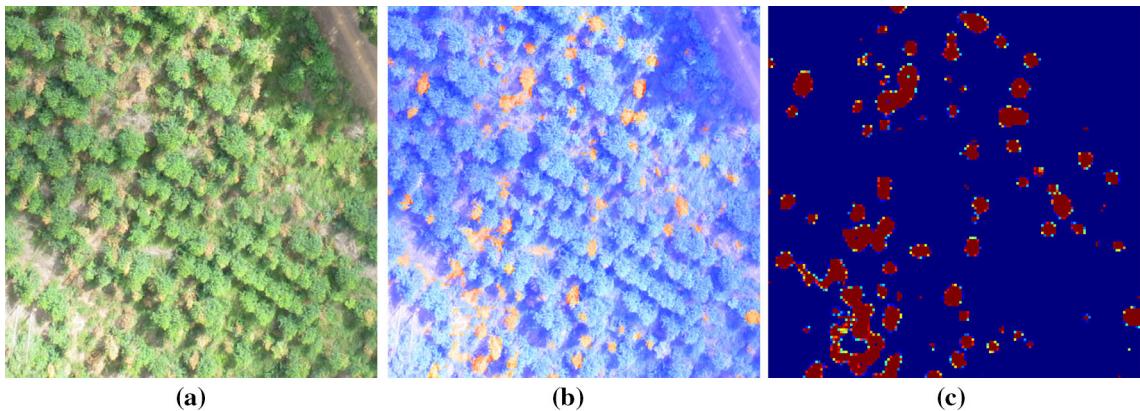
### 5.1 Evaluation of logistic regression model

As stated in Sect. 3.2.2, we manually defined 32 features per class (Table 1). The Contextual Block methodology doubles the number of features, as there are 32 features from the Internal Block and 32 features from the Contextual Block, summing up to 64 image descriptors. Figure 8 shows a classified image and the probability output from the LR model (colors range from blue, which represents healthy trees and other structures, to red, which represents diseased trees).

The dataset was split into ten images for training and five for testing, which led to 10,993 samples of diseased trees and 73,031 samples of healthy trees and other structures. The sizes of the Contextual Block and Internal Block were changed for the selection of the best model for the classification module. Internal block values were chosen from  $4 \times 4$  and  $8 \times 8$  pixels and Contextual blocks were  $10 \times 10$ ,  $20 \times 20$  and  $30 \times 30$  pixels. Evaluation was performed by comparing pixels from the ground-truth and the classified image. Table 2 shows the F-score for each block size to the LR model. We verified that smaller internal blocks ( $4 \times 4$  pixels) leads to models with better scores when allied to a contextual block size of  $30 \times 30$  pixels. Our intuition for this behaviour is that smaller internal blocks reach essentially only the treetops, while larger blocks take information that is not just from the canopies, but soil and other nearby objects, so smaller blocks tend to have a better classification score.

### 5.2 Route planning module

BO, SA, RRT, Random Points and Grid Trajectory methods were evaluated as potential candidates for this module. Each method provides a global route and search for news points of destination starting from the source of the UAV up to a distance of 2000 m (estimated battery autonomy time).



**Fig. 8** Classification of images captured by the UAV. Red pixels represent diseased trees and blue pixels represent healthy trees/other structures. **a** Image captured by UAV, **b** classified image, **c** classification probability from the Logistic Regression classifier

**Table 2** F-score for each contextual block size

Contextual block × Internal block					
10 × 4	20 × 4	30 × 4	10 × 8	20 × 8	30 × 8
0.75	0.81	<b>0.85</b>	0.71	0.79	0.82

Bold value indicates the best result relative to all other comparisons

As a baseline, the Grid Trajectory method covers the entire area adopting a predefined route and its estimated distance is around 1000 m. The Random Points method raffles random destination points without a heuristic function. For all methods, each point represents an image captured by the UAV that is collected during the execution of the route.

Continuous BO (CBO) also provides route planning between intermediate points (the local route). Therefore, it considers uncertainty along the path from one point to another for a better selection of each destination. The BO parameters for path planning were selected empirically and defined in all experiments to be  $\sigma_v^2 = 100$  and  $\sigma_l^2 = 0.02$ . The SA considers only the uncertainty of the destination node, which is the result of the global route. The RRT follows a method similar to the CBO and initially defines a destination point and then a route, considering the uncertainty along the way. In RRT, whenever the UAV takes a picture, the captured area is labeled as visited and considered an obstacle, so the UAV avoids the region except when it is inevitable (i.e. when it must visit a closed region surrounded by obstacles).

We analyzed the BO and SA approaches for route planning and adopted the RRT algorithm to provide the path planning, due to its performance achieved in previous studies (Vivaldini et al. 2016). Two tests were performed combining the RRT algorithm. These tests are described below:

1. BO chooses the goal point and RRT suggests the path.
2. SA chooses the goal points and RRT suggests the path.

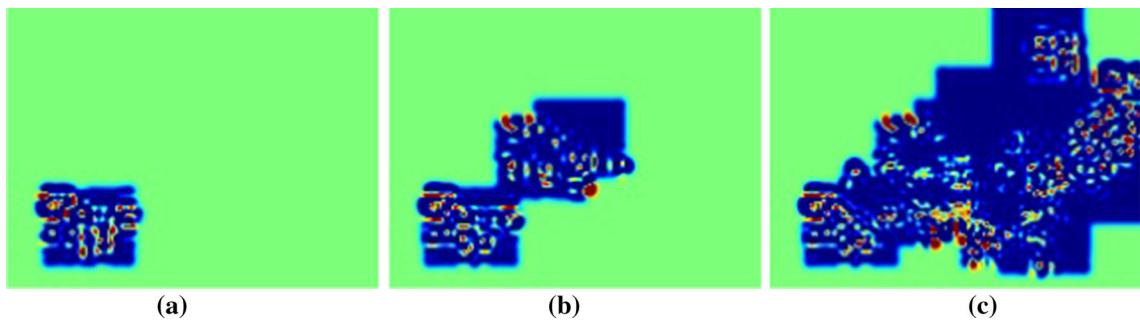
We also tested the Offline CBO method, which calculates all points for path planning prior to the execution of the route. The algorithm was implemented to validate the efficiency of BO, even without the knowledge necessary to choose the next point. It is also the standard approach for online applications, since most UAVs lack the computational power necessary for the real-time processing of available information. The results of the comparisons are shown in the following sections.

### 5.2.1 Active classification

In each scenario, the GP map was initialized with values of 0.5, which indicates an unknown environment. The values were then updated by the active classification results during UAV flight. We considered the classification values of 0 to 0.3 as diseased trees (red),  $0.3 \leq p(X) \leq 0.7$  as uncertain areas (green) and 0.7 to 1 as healthy trees/other structures (blue) (Fig. 9). The trajectory and final map classification after the completion of each route planning method for scenario 1, when using different techniques, is shown in Fig. 10.

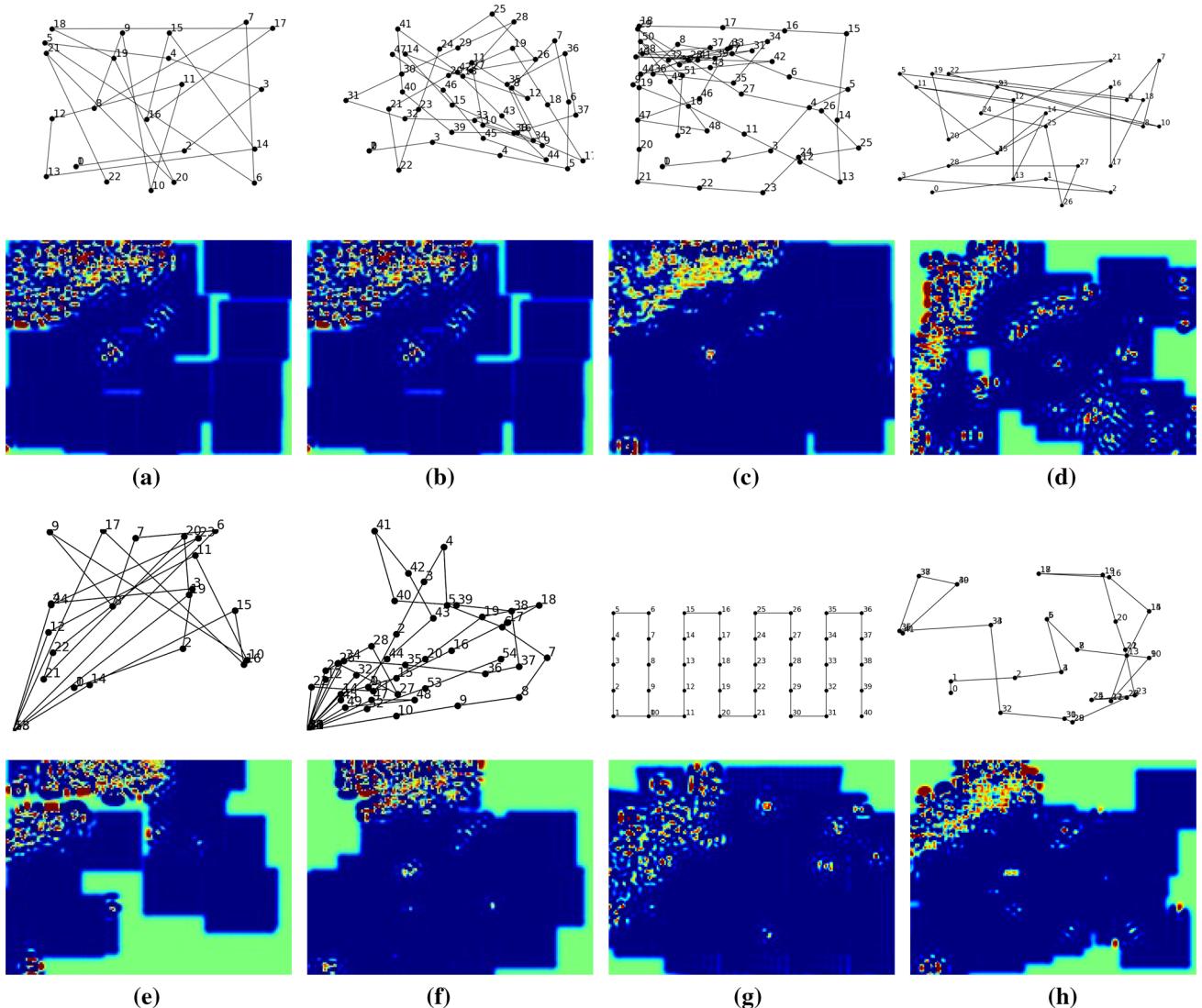
### 5.2.2 Uncertainty decrease over distance

Five scenarios were used, and for each one we analyzed different route planning methods comparing distance and uncertainty over the map area (generated by LR model and GP interpolation). Figure 11 shows the decrease of uncertainty over distances on the map. As the vehicle navigates in an initially unknown environment, the results validate the similar nature of the methods adopted, since they define both destination points and a route considering uncertainty along the path. As expected, all methods decrease values of uncertainty as new points are added. The challenge is then to reliably decrease the values within a smaller travelled distance.



**Fig. 9** Classification estimates during navigation. Diseased trees are depicted in red, healthy trees/other structures are depicted in blue and uncertain areas are depicted in green. **a** Initial map (first image), **b** map

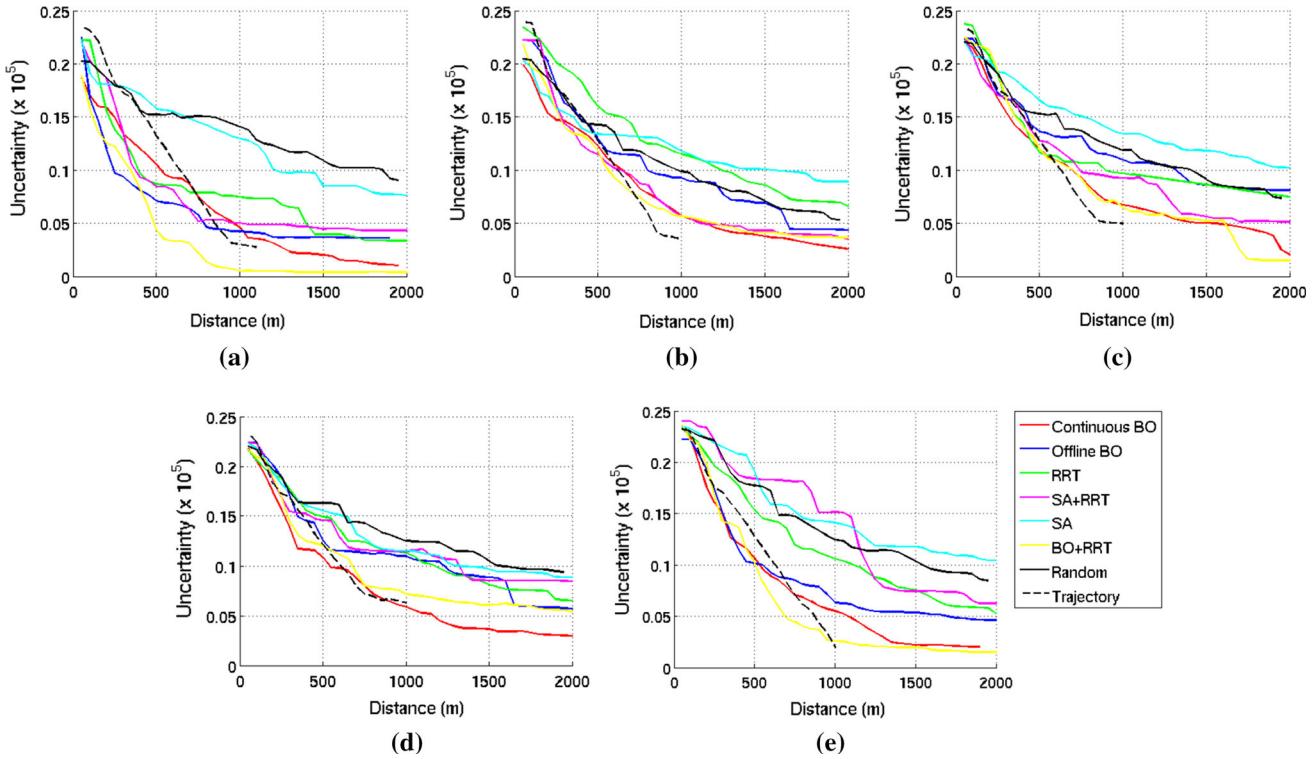
after navigation to the first sample point, **c** intermediary map, after navigation to six sample points (Color figure online)



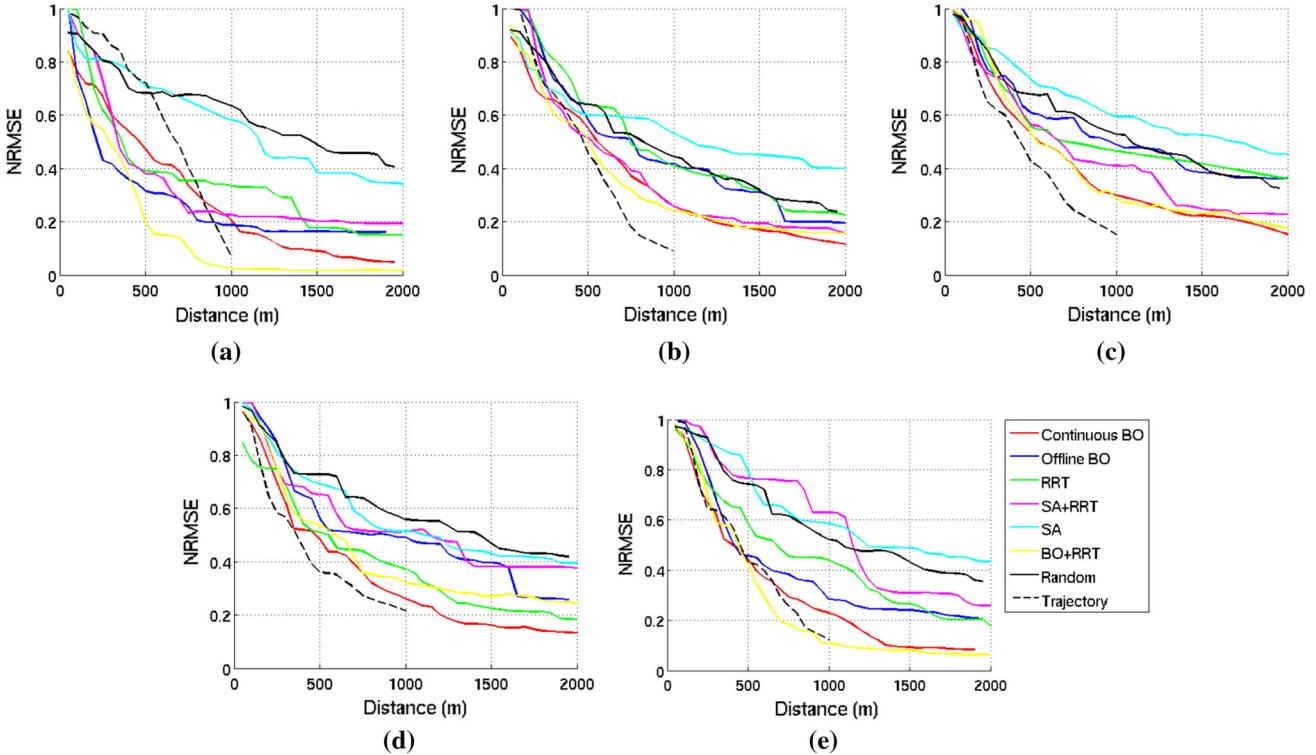
**Fig. 10** Trajectory and coverage results using different route planning techniques. The top row depicts points sampled during navigation, and the bottom row shows the resulting map after navigation. **a** CBO, **b** offline CBO, **c** BO + RRT, **d** random, **e** SA, **f** SA + RRT, **g** trajectory, **h** RRT

For scenarios 1, 2, 3 and 4, CBO and BO + RRT have the same end result at the end of the flight in 2000 m, but CBO

reaches its lowest values in approximately 1800 m, whereas BO + RRT reaches its smallest uncertainty values between



**Fig. 11** Uncertainty decrease over distance. **a** Scenario 1, **b** Scenario 2, **c** Scenario 3, **d** Scenario 4, **e** Scenario 5



**Fig. 12** Normalized root mean square error (NRMSE) decrease over distance. **a** Scenario 1, **b** Scenario 2, **c** Scenario 3, **d** Scenario 4, **e** Scenario 5

800 and 1000 m and ensures a total coverage of each scenario (i.e. Fig. 11). The grid trajectory demonstrates the smallest

number of uncertain areas, because its path guarantees a good coverage of the scenario. The random method provided a

worse result for not having a metric from which to choose the path. SA showed the worst performance and required a longer distance to travel the entire map. On the other hand, when SA + RRT was adopted, the results improved significantly and were similar to those of Offline CBO.

A well-known problem of SA is the local minimum, since we cannot guarantee that it has found an optimal solution after convergence. Thus, a complementary method is necessary for this purpose, in our case the RRT. Also, CBO and Offline CBO introduce extra information about the problem, in the form of mean and variance values for uncertainty in ambiguous areas. Furthermore, the BO framework has an acquisition function, that is used to minimize these ambiguous regions (non-classified) in a structured manner, thus producing a more reliable map of the environment.

### 5.2.3 NRMSE decrease over distance

The predicted probabilities in the GP map and the reference image (classified by a geologist and serving as ground truth for the purposes of training and validating) were compared using the Normalized Root-Mean-Square Error (NRMSE) metric. This metric quantifies how similar the predicted image is to the reference image, ranging from 1 (completely different) to 0 (same image). Therefore, the estimated probabilistic aspect provided by LR and interpolated with GP was not discarded. Figure 12 shows a faster decrease of NRMSE when using CBO-based approaches in comparison to SA-based methods, as its routes are more informative. In Scenarios 3 and 5, the result of CBO is very close to that of BO + RRT, while BO + RRT provided the best results in Scenarios 1, 2 and 4. SA obtained the worst final error, as it tends to produce a substantial amount of overlapping between its images. The Grid Trajectory provides a lower final error for the entire area, due the presence of non-overlapping images. In general, NRMSE from Offline CBO was better than SA + RRT, because the latter contained more areas without classification at the end of the mission.

## 6 Conclusion

This paper proposes a novel route planning algorithm for active classification using UAVs, aiming to maximize collected information within a given distance, determined by vehicle flight autonomy. Two techniques, namely CBO and SA, were evaluated by themselves and with the addition of RRT, providing both way-points for navigation and the trajectory between them. A LR classifier was used to classify the image frames collected, searching for diseased trees, and a GP was used to interpolate this information, producing a navigational map. In contrast to traditional methods of route planning, which have pre-established targets, the proposed

active classification technique can adapt to the constant flow of new information. From its current position, it uses CBO to search for the best destination point to be visited given its current goal (i.e., exploration or exploitation), to maximize the amount of information collected at each step. From these points, an RRT algorithm identifies trajectories considered relevant to be traversed.

The main advantage of the proposed BO + RRT framework is the combination of route and path planning with active classification. The BO algorithm selects the best destination points from its current incomplete environment model, while RRT calculates the best path to be taken to reach the next destination point. During the execution of the path, active classification from LR is applied to update the current environment as new data is incorporated, decreasing overall uncertainty. The proposed framework can be applied to a wide variety of different scenarios, in which we have an established target but do not know its location in an unknown environment. The vehicle can incrementally learn an incomplete model of the environment and use this information to constantly adapt its navigation path.

As future work, the authors plan to employ this methodology in an online scenario using embedded systems, which would enable onboard processing in the UAV itself. The introduction of different sensors (i.e. multi-spectral and thermal cameras) would also greatly increase the amount of information available for the classification algorithm, both increasing the accuracy of generated maps and allowing the detection of a wider variety of pattern. Additionally, by promoting changes in altitude during flight it would be possible to generate multiple resolution maps, with lower resolutions producing a larger field of view for faster initial surveys and higher resolutions producing more detailed representations for better classification accuracy.

**Acknowledgements** The authors acknowledge by the CNPq Foundation (process 400699/2016-8), CAPES and FAPESP for the financial support. This research project was also supported by funding from the Faculty of Engineering & Information Technologies, The University of Sydney, under the Faculty Research Cluster Program. Lastly, Federal University of Uberlândia. Funding was provided by CNPq - Brazil (Grant Nos. 400.395/2014-2).

## References

- Albore, A., Peyrard, N., Sabbadin, R., & Teichteil-Knigsbuch, F. (2015a). Extending an online (re) planning platform for crop mapping with autonomous UAVs through a robotic execution framework. In *Proceedings of ICAPS 2015 scheduling and planning applications workshop (SPARK)*.
- Albore, A., Peyrard, N., Sabbadin, R., & Teichteil-Knigsbuch, F. (2015b). An online replanning approach for crop fields mapping with autonomous UAVs. In *Proceedings of the twenty-fifth international conference on automated planning and scheduling, Jerusalem, Israel*.

- Bedendo, I. P. (1995). *Doenças vasculares Manual de Fitopatologia: Princípios e Conceitos*. São Paulo: Agronômica Ceres.
- Bernardini, S., Fox, M., & Long, D. (2014). Planning the behaviour of low-cost quadcopters for surveillance missions. In *Proceedings of international conference on automated planning and scheduling, Portsmouth, USA*.
- Candiago, S., Remondino, F., de Giglio, M., Dubbini, M., & Gatelli, M. (2015). Evaluating multispectral images and vegetation indices for precision farming applications from UAV images. *Remote Sensing*, 7, 4026–4047.
- Dalamagkidis, K., Valavanis, K. P., & Piegl, L. A. (2012). *On integrating unmanned aircraft systems into the national airspace system into the national airspace system* (2nd ed.). Berlin: Springer. ISBN 978-94-007-2478-5.
- Degroote, A., Koch, P., & Lacroix, S. (2016). Integrating realistic simulation engines within the Morse framework. In *2016 IEEE/RSJ international conference on intelligent robots and systems, Daegu, Korea*.
- Donald, B., Xavier, P., Canny, J., & Reif, J. (1993). Kinodynamic motion planning. *Journal of the ACM*, 40(5), 1048–1066.
- Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., & Gharamani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. In *Proceedings of the international conference on machine learning*.
- Echeverria, G., Lassabe, N., Degroote, A., & Lemaignan, S. (2011). Modular open robots simulation engine: Morse. In *2011 IEEE international conference on robotics and automation (ICRA)* (pp. 46–51). IEEE.
- Echeverria, G., Lemaignan, S., Degroote, A., Lacroix, S., & Karg, M. (2012). Simulating complex robotic scenarios with Morse. In *3rd international conference on simulation, modeling, and programming for autonomous robots, Tsukuba, Japan*.
- Egerstedt, M., & Martin, C. F. (2001). Optimal trajectory planning and smoothing splines. *Automatica*, 37, 1057–1064.
- Engelbrecht, A. P. (2006). *Fundamentals of computational swarm intelligence*. London: Wiley.
- FAA. (2016). *Unmanned aircraft systems*. Washington: Federal Aviation Administration.
- Ghamry, K. A., Kamel, M. A., & Zhang, Y. (2016). Cooperative forest monitoring and fire detection using a team of UAVS–UGVs. In *International conference on unmanned aircraft systems (ICUAS)*.
- Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing*. Upper Saddle River, NJ: Prentice Hall.
- Grocholsky, B., Keller, J., Kumar, V., & Pappas, G. (2006). Cooperative air and ground surveillance: A scalable approach to the detection and localization of targets by a network of UAVs and UGVs. *IEEE Robotics & Automation Magazine*, 13, 16–26.
- Hensman, J., Fusi, N., & Lawrence, N. D. (2013). *Gaussian processes for big data*.
- Ho, Y., & Liu, J. (2010). Simulated annealing based algorithm for smooth robot path planning with different kinematic constraints. In *ACM symposium on applied computing, Sierre, Switzerland*.
- Hyttinen, E., Krägic, D., & Detry, R. (2015). Learning the tactile signatures of prototypical object parts for robust part-based grasping of novel objects. In *IEEE international conference on robotics and automation*.
- Ingber, L., & Rosen, B. (1992). Genetic algorithms and very fast simulated reannealing: A comparison. *Mathematical and Computer Modelling*, 16, 87–100.
- Jensen, J. R. (2007). *Remote sensing of the environment: An earth resource perspective*. Upper Saddle River, NJ: Pearson Prentice Hall. ISBN-10: 0131889508.
- Karakaya, M. (2014). UAV route planning for maximum target coverage. *International Journal of Computer Science and Engineering*, 4(1), <https://doi.org/10.5121/cseij.2014.4103>.
- Kim, S. J., Lim, G. J., Cho, J., & Côté, M. J. (2017). Drone-aided healthcare services for patients with chronic diseases in rural areas. *Journal of Intelligent and Robotic Systems*, 88, 163–180.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.
- Lavalle, M., & Kuffner, S. J. J. (2000). Rapidly-exploring random trees: Progress and prospects. In *Proceedings of workshop on the algorithmic foundations of robotics, San Francisco*.
- Lemaignan, S., Hanheide, M., Karg, M., Khambaita, H., Kunze, L., Lier, F., et al. (2014). *Simulation and HRI recent perspectives with the MORSE simulator* (pp. 13–24). Cham: Springer.
- Liu, Y., Zhong, Y., Chen, X., Wang, P., Lu, H., Xiao, J., & Zhang, H. (2016). The design of a fully autonomous robot system for urban search and rescue. In *IEEE international conference on information and automation (ICIA)*.
- Ludington, B., Johnson, E., & Vachtsevanos, G. (2006). Augmenting UAV autonomy: Vision-based navigation and target tracking for unmanned aerial vehicles. *IEEE Robotics & Automation Magazine*, 13, 63–71.
- MAPA. (2015). Ministry of Agriculture, Livestock and Food Supply. a, 1:1.
- Marchant, R., & Ramos, F. (2012). Bayesian optimisation for intelligent environmental monitoring. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2242–2249).
- Medeiro, F. L. L., & da Silva, J. D. S. (2010). A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation. *Advances in Artificial Intelligence, Lecture Notes in Computer Science*, 6404, 213–22.
- Meng, H., & Xin, G. (2010). UAV route planning based on the genetic simulated annealing algorithm. In *International conference on mechatronics and automation, Xi'an, China*.
- Milliez, G., Ferreira, E., Fiore, M., Alami, R., & Lefèvre, F. (2014). Simulating human–robot interactions for dialogue strategy learning. In *International conference on simulation, modeling, and programming for autonomous robots* (pp. 62–73). Berlin: Springer.
- Mulgaonkar, Y., & Kumar, V. (2014). Autonomous charging to enable long-endurance missions for small aerial robots. In *Proceedings of micro and nanotechnology sensors, systems, and applications VI, Baltimore, United States*.
- Negro, D. R., Junior, T. A. F. S., Passos, J. R. S., Sansgolo, C. A., Minhoní, M. T. A., & Furtado, E. L. (2014). Biodegradation of eucalyptus urograndis wood by fungi. *International Biodeterioration & Biodegradation*, 89, 95–102.
- Ng, A. Y. (2004). Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on machine learning, ICML '04, New York, NY, USA*. New York: ACM.
- Ojala, T., Pietikainen, M., & Maenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with LBP. *TPAMI*, 24, 971–987.
- Park, H., Lee, B. H. Y., & Morrison, J. R. (2017). Persistent UAV security presence service: Architecture and prototype implementation. In *2017 international conference on unmanned aircraft systems (ICUAS)* (pp. 1800–1807).
- Pérez-Ortíz, M., Gutiérrez, P. A., Peña, J. M., Torres-Sánchez, J., López-Granados, F., & Hervás-Martínez, C. (2016). Machine learning paradigms for weed mapping via unmanned aerial vehicles. In *2016 IEEE symposium series on computational intelligence (SSCI)* (pp. 1–8).
- Ponti, M., Chaves, A. A., Jorge, F. R., Costa, G. B. P., Colturato, A., & Branco, K. R. L. J. C. (2016). Precision agriculture: Using low-cost systems to acquire low-altitude images. *IEEE Computer Graphics and Applications*, 36(4), 14–20.
- Popović, M., Hitz, G., Nieto, J., Sa, I., Siegwart, R., & Galceran, E. (2017). Online informative path planning for active classification

- using UAVs. In *2017 IEEE international conference on robotics and automation (ICRA)* (pp. 5753–5758).
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: An open-source robot operating system. In *ICRA workshop on open source software*.
- Rasmussen, C. E., & Williams, K. I. (2006). *Gaussian processes for machine learning*. Cambridge: MIT Press.
- Reid, A., Ramos, F., & Sukkarieh, S. (2011). Multi-class classification of vegetation in natural environments using an unmanned aerial system. In *2011—IEEE international conference on robotics and automation (ICRA), Shanghai, China*.
- Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Proceedings of the 18th International Conference on Neural Information Processing Systems* (pp. 1257–1264).
- Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems* (pp. 2951–2959).
- Souza, J. R., Mendes, C. C. T., Guizilini, V., Vivaldini, K. C. T., Coltrato, A., Ramos, F., & Wolf, D. F. (2015). Automatic detection of ceratocystis wilt in eucalyptus crops from aerial images. In *2015 IEEE international conference on robotics and automation (ICRA)* (pp. 3443–3448).
- Stoer, J., Bulirsch, R., Bartels, R. H., Gautschi, W., & Witzgall, C. (2002). *Introduction to numerical analysis. Texts in Applied Mathematics*. New York: Springer.
- Tai, L., Li, S., & Liu, M. (2017). *Autonomous exploration of mobile robots through deep neural networks* (pp. 1–9).
- Turker, T., Sahingoz, O. K., Springer, Yilmaz, G. (2015). 2D path planning for UAVs in radar threatening environment using simulated annealing algorithm. In *International conference on unmanned aircraft systems, Denver, CO, USA*.
- Vivaldini, K. C. T., Guizilini, V., Oliveira, M. D. C., Martinelli, T. H., F.Ramos, & Wolf, D. F. (2016). Route planning for active classification with UAVs. In *2016—IEEE international conference on robotics and automation (ICRA), Stockholm, Sweden*.
- Weinstein, A. L., & Schumacher, C. (2007). *UAV scheduling via the vehicle routing problem with time windows* (p. 17).
- Witwicki, S., Castillo, J. C., Messias, J., Capitan, J., Melo, F. S., Lima, P. U., & Veloso, M. (2017). *Autonomous surveillance robots: A decision-making framework for networked multiagent systems* (pp. 52–64).
- Yang, K., Gan, S. K., & Sukkarieh, A. (2013). Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with an UAV. *Advanced Robotics*, 27, 431–443.
- Zhou, Z. G., Zhang, Y. A., & Zhou, D. (2016). Geometric modeling and control for the full-actuated aerial manipulating system. In *2016 35th Chinese control conference (CCC)* (pp. 6178–6182).

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Kelen C. T. Vivaldini automation and intelligent systems.

**Kelen C. T. Vivaldini** is an Assistant Professor in the Department of Computer Science at Federal University of São Carlos (UFS-Car), São Carlos, SP, Brazil. She received her B.Sc. in Computer Engineering from the UNORP in 2007, M.Sc. and Ph.D. degree in Mechatronics Engineering at the University of São Paulo (USP), in 2010 and 2014. A current post-doctoral fellow in Computer Science at the University of São Paulo (USP). She research interests include robotics, route planning,



**Thiago H. Martinelli** received the Aeronautical Engineering degree from the University of São Paulo (USP), São Carlos, Brazil in 2006 and the M.Sc. degrees in Computer Science from the same university, in 2013. During his M.Sc., he developed a video stream adaptation system to images captured by UAV. Currently, he is a Ph.D. student in Computer Science at the University of São Paulo (USP), developing a map building system based in lifelong learning algorithms. Two years of experience

working in CAD software development and maintenance in Embraer. Two years working as UAV designer in the company Xmobots. Two years working as lecturer of programming techniques, linear algebra and embedded systems, all for engineering undergraduate courses.



**Vitor C. Guizilini** obtained his Ph.D. at the University of Sydney in 2014, and is now a Research Fellow with the University of Sydney, School of Information Technologies, and a member of the Human-Centred Technology (HCT) group. His main research interests are Computer Vision, Machine Learning, Bayesian Systems and Data Fusion.



**Jefferson R. Souza** is an Assistant Professor in the Department of Information Systems at Faculty of Computing in the Federal University of Uberlândia (Monte Carmelo/MG). He obtained his Ph.D. degree in Computer Science at the University of São Paulo (USP) in 2014. Furthermore, he did the postdoctoral at USP in 2014. Currently, he is working with Unmanned Aerial Vehicles and Autonomous Car. Collaborates with the Mobile Robotics Lab at USP and Robotics Learning and Reasoning Group at The University of Sydney. His research interests are Mobile Robots, Machine Learning and Computer Vision.



**Denis F. Wolf** is an Associate Professor in the Department of Computer Systems at the University of São Paulo (USP). He obtained his Ph.D. degree in Computer Science at the University of Southern California USC in 2006. Currently he is Director of the Mobile Robotics Laboratory at USP and member of the Center for Robotics at USP. His current research interests are Mobile Robotics, Intelligent Transportation Systems, Machine Learning and Computer Vision.



**Matheus D. Oliveira** is a Data Scientist at the innovation department of B2W Digital. He obtained his M.Sc. degree in Computer Science at the University of São Paulo (USP) in 2016, doing research in the area of disease detection in eucalyptus plantations using machine learning. His research interests are Data Science, Machine Learning and Deep Learning.



**Fabio T. Ramos** is an Associate Professor in machine learning and robotics at the School of Information Technologies, University of Sydney. He received the B.Sc. and the M.Sc. degrees in Mechatronics Engineering at University of São Paulo, Brazil, in 2001 and 2003 respectively, and the Ph.D. degree at University of Sydney, Australia, in 2007. From 2007 to 2010 he was an Australian Research Council research fellow at the Australian Centre for Field Robotics (ACFR) and ARC

DECRA Fellow from 2012 to 2014. His research focuses on statistical machine learning techniques for large-scale data fusion with applications in robotics, mining, environmental monitoring and healthcare.