1. Executive Summary
   1.1 Summary of deliverable content and initial objectives
   1.2 Partners involved
   1.3 Relation with other work packages and tasks
   1.4 Weblinks to videos, flyers,...
   1.5 Dissemination / IPR policy

2 Introduction

3 Implementation with 2D segmentation
   3.1 Plant detection
   3.2 Path planning
   3.3 Exploratory experiments in outdoor environments

4 Improving the 2D segmentation via learning
   4.1 Building the dataset

5 Deep learning approach to semantic segmentation

6 Improving segmentation using the 3D structure
   6.1 3D reconstruction of the workspace
   6.2 Detection of plants and projection of the 3D point cloud

7 Conclusion
1. Executive Summary

1.1. Summary of deliverable content and initial objectives

This deliverable is about the computer vision algorithms used for the weeding application. We start by describing the algorithmic pipeline currently used with a 2D segmentation of plants using a threshold-based method. We then discuss the limitations of these methods and possible ways to overcome it using learning-based algorithms and 3D data.

1.2. Partners involved

Leader: CNRS
Participants: Sony CSL

1.3. Relation with other work packages and tasks

Relation to the tasks of WP5: This deliverable is related to task “T5.1 Sensors selection and data acquisition rate”. As we discuss the comparison of 2D and 3D data and it is also related to “T5.3 3D+t plant segmentation”.

Relation to other work packages in ROMI: The hardware aspects of the robot for weeding and the integration of the software are the subject of the work package WP2. The test of the weeding application are detailed in work packages WP7 and WP2.

1.4. Weblinks to videos, flyers,...

The annotated dataset for semantic segmentation can be found here.

1.5. Dissemination / IPR policy

Early version of this work was presented at the European forum dedicated to the future use of ICT in the agri-food sector, bioresource and biomass sector (EFITA in Montpellier, 2017). It was attributed the best poster award and the related publication can be found here. This work was also presented in the European Robotics Forum (ERF in Tampere, 2018).

This document is available under the Creative Commons BY-CC 4.0 International license.
2. Introduction

Among the applications of agriculture robotics, weed control is a key issue: it is critical to avoid reductions in crop productivity. In the strategies to limit weed proliferation, herbicides are expensive and generate pollution while practices in organic farming, mostly manual, are time-consuming and physically challenging. In both conventional and organic farming, weeding robots are thus appealing innovations and it has been proposed that they could operate by several methods: chemical, electrical, thermal or mechanical perturbations of the weeds.

Considering ROMI’s commitment for organic farming and sustainability, we designed a weeding robot based on a simple mechanical weeding, involving no chemicals. A major challenge for such a robot is to discriminate crop from weeds in culture beds. Underlying methods should first involve collecting different data (with appropriate sensors) such as color, hyperspectral reflection, shape or location cues. Second, large annotated database would then be required to train classifiers.

Here, we present a first solution to this challenge: preventive weeding and simple “plant versus soil” discrimination. This solution applies to situations where the culture bed is initially free of weeds: young plants are grown in a greenhouse and then planted out in the prepared and weed-free culture beds. This is a common practice for many species cultured in market farms. The weeding robot is mounted on the LettuceThink rover platform and consists in a CNC machine augmented with a rotating hoe that is moved through the working area underneath the robot. The task is to move the hoe along a path that covers the ground in the working area while avoiding the plants. Before plants are transferred to the bed, the robot maintain the area clean of plants. After planting, the weeding process is straightforward and can be executed in 3 steps:

- Detect regions of the workspace occupied by plants of interest.
- Generate a toolpath covering the workspace except for the regions occupied by plants.
- Run the tool through the generated path.

Our hypothesis is that a regular application of this process prevents most of the weeds from developing. We show here the results from the exploratory experiments we conducted during this first year.

3. Implementation with 2D segmentation

3.1. Plant detection

A large collection of color indices have been proposed for the detection of plants. The most robust index was selected by testing on images (see Fig. 1) from a wide variety of external conditions (with different lighting conditions and types of soil). The index used in practice is an excess green index, as described in

---

Hamuda E et al. (2016) with slight difference in the normalization. It is based on the rescaled values of the image channels, at pixel \((i, j)\):

\[
\overline{R}_{ij} = \frac{R_{ij}}{\max(R)}, \quad \overline{G}_{ij} = \frac{G_{ij}}{\max(G)}, \quad \overline{B}_{ij} = \frac{B_{ij}}{\max(B)}
\]

where \(R\), \(G\) and \(B\) are the matrices corresponding to red, green and blue channels of the image.

The excess green index is then defined, at pixel \((i, j)\), as:

\[
C_{ij} = \frac{3 \cdot \overline{G}_{ij}}{\overline{R}_{ij} + \overline{G}_{ij} + \overline{B}_{ij}} - 1. \]

Figure 1: Comparison of common color indices, detailed in Hamuda E, Glavin M & Jones E (2016), for the computation of the plant masks. Several of the indices have similar performances on most images. (First line) Color index computed on a test image. (Second line) Mask obtained after applying a threshold on the index. (Third line) Result of filtering the small patches using morphological operations.

\footnote{Hamuda E., Glavin M. & Jones E. (2016) 'A survey of image processing techniques for plant extraction and segmentation in the field.' Computers and Electronics in Agriculture, vol.125, pp. 184-199}
On images of young plants on ground, the histogram for this index is bimodal with one main peak for the ground and a secondary peak for the plants. The clustering of the pixels in an image can be performed using K-means or Gaussian mixture models to determine the threshold for the separation of the part of the histogram related to the ground and the one related to plants (an implementation of automatic thresholding using Otsu’s method available in the OpenCV library performs well on most images).

Figure 2: Weeding application algorithms: (Left) Excess green color index and mask defining the domains occupied by plants. (Right) Tool path generated using the modified boustrophedon algorithm.

The thresholded image still contains small patches of pixels with high excess green index which are not the cultivated crops. Those may either be plants of small size (weeds) or outlier pixels from the ground appearing as green. To ensure that the mask describing the regions occupied by plants of interest does not include those pixels, successive erosions and dilations are applied. This acts as a size filter of segmented regions and the parameters of these morphological operations (size and shape of the kernel and number of iterations) are used to choose the minimum size of the regions identified as plants (see Fig. 1).

3.2. Path planning

Based on the mask of the segmented plants, the area to be avoided by the center of the hoe is taken as all points whose distance to the plant area is less than the radius of the tool. The optimal solution to the two dimensional covering path problem on a convex domain with no obstacles is a simple zig-zag path, commonly known as the boustrophedon\(^3\).

As a first algorithm for path generation, the boustrophedon path for the workspace with forbidden areas is modified so that parts of the path that go through forbidden regions are substituted with valid regions. At each portion of the boustrophedon which crosses a forbidden region, entry and exit points

are identified. To have pixel accuracy in the detection of those points, both the initial boustrophedon path and forbidden areas contours are densified using linear interpolation. For each pair of entry/exit points, the shortest path along the border of the corresponding forbidden zone connecting those points is substituted to the straight line of the original boustrophedon. Finally, the number of points composing the path is reduced using the Douglas-Peucker algorithm.\(^4\)

**Figure 3:** Comparison of paths generated by the modified boustrophedon (yellow) and the elastic net (purple): (Left) Sample paths when plants are small. (Right) Path length depending on the size of plants (simulated through iterated dilation of the small plants). Curves crossing suggest a plant size for which elastic net generates shorter paths, but at the cost of longer computation time.

The modified boustrophedon is well suited for young plants, but as plants grow, although the ground region to be covered by the tool shrinks, the resulting path length increases (see Fig. 3) due to multiple passages on the contours of the forbidden regions. As an alternative to this simple algorithm, a more computationally demanding algorithm is designed using the following steps:

- Generate cells of size comparable with the tool, covering the ground regions,
- Find the shortest path passing through all centers of the cells once and only once,
- Reduce the number of points along the path using the Douglas-Peucker algorithm.

There are several possible algorithms to generate ground cells, e.g. with the SLIC superpixel algorithm.\(^5\) The second step can be formulated as a traveling salesman problem (TSP) and there are also multiple methods available to solve it. Elastic net is a kind of self-organizing map which reaches approximate solutions with good performance when the problem dimension is reasonable\(^6\) and we found that it performs well on this problem. The centers of superpixels \(S_i\) with \(i \in \{1 \cdots N\}\) are associated with cities in the TSP so that the shortest path along those points will cover the ground. Starting with a

\(^4\) David H Douglas et Thomas K. Peucker, « Algorithms for the reduction of the number of points required to represent a digitized line or its caricature », *Cartographica: The International Journal for Geographic Information and Geovisualization*, University of Toronto Press, vol. 10, n° 2, 1973, p. 112-122


number of nodes (Pj with j ∈ [1 · · · M]) denser than the number of cities (N < M) and organized along a path, we solve the optimization problem with a cost function comprising 2 terms:

- $C_1(P_j, K) = -\alpha K \sum_i \log(\sum_j e^{-\frac{d_{ij}^2}{2\sigma^2}})$ is the term accounting for the path passing through the centers of cells modelled as a attraction potential by centers of cells on path nodes.
- $C_2(P_j) = \beta \sum_j (P_j - P_{j+1})^2$ is the term accounting for the path length modelled as an elastic interaction among path nodes.

The update rule for moving path nodes is then:

$$\Delta P_j = \alpha \sum_i w_{ij} (S_i - P_j) + \beta K (P_{j+1} + P_{j-1} - 2P_j),$$

with:

$$w_{ij} = \frac{e^{-\sigma^2 d_{ij}^2}}{\sum_k e^{-\sigma^2 d_{ik}^2}}.$$

Here the parameter K is a regularizing factor, varied along iterations of the algorithm, so that the initial steps are not trapped in a local minimum. In the first iterations of the algorithm, K is large and each path node feels the potential of several centers around and as the algorithm goes on interactions are frozen so that only interactions with closest centers remain.

We tested path generation with both modified boustrophedon and elastic net on a workspace including plants at various (simulated) growth stages. The path is shorter with the modified boustrophedon at early stages showing that the elastic net solution is not globally optimal. As plants grow, the elastic net solution gets shorter than the modified boustrophedon, reflecting the inefficiencies mentioned above for the modified boustrophedon. Paths obtained with the elastic net are decreasing in length as plants grow, as expected, and at some stage of the growth, paths generated by the elastic net are shorter than the ones obtained by the modified boustrophedon. At this point, elastic net solutions are better although at a higher computational cost. Another possible algorithm, yet to be tested, is based on a cellular decomposition of the ground domains and running a boustrophedon path on each cell.

### 3.3. Exploratory experiments in outdoor environments

We are testing the weeding application on radishes cultivated in plain ground. We continue to refine the application to account for unexpected aspects of an open-field experiment. For example, red markers were added on the field to make the identification of the workspace easier on images. One of the challenges that we encountered is the high contrast in the image generated by shade, which may result in plants or markers detection to fail. As plants grow, markers may be hidden by plants or no ground may be left to hoe. Although weeding may be unnecessary when plants are big since the growth of weeds is slowed down due to foliage shading, the path generation algorithm could be adapted to generate a covering path of the connected components of the ground so that it could be used when

---

plants are overlapping, leaving isolated islands of ground to be hoed. We also noticed that, for radish, when the plant grows, leaves are few centimeters above the ground so that using the contour of leaves as the border for the forbidden zone may leave a large portion of the ground untouched. In this case, it may be better to take finer details of the plant architecture, like the 3D geometry, into account in the definition of the forbidden zone.

**Figure 4:** Test of the weeding application on radishes cultivated in open-field: (Left) The hoe is passed before transplanting and every week afterwards. (Middle) The hoe is passed before transplanting only. (Right) The ground is left untouched.

Preliminary evaluation shows that a pass every week is enough to maintain a low density of weeds compared to a bed where the hoe was not used during plant growth and that had many more weeds. In both case, the ground was hoed before transplanting the radishes. A single pass before transplanting is quite efficient in delaying the growth of weeds by about a week. Most of the tests with the weeding tool have been performed on beds without cultivars and more details and images on these experiments are available in [deliverable 7.2](#).

### 4. Improving the 2D segmentation via learning

Although we showed that useful segmentation masks may be obtained from applying a threshold to a color index, this method suffers from inherent limitations. An assumption in using a threshold-based method is that the distribution of color index is bimodal (see Fig. 2). In the data we collected, we observed that this assumption is not always verified. Shadows and reflections of the sunlight, for example, are detrimental to the detection of plants. In an example illustrated in Figure 5, the distribution of color indexes has three modes resulting in difficulties for choosing the threshold using Otsu’s method or a Gaussian mixture model.

We studied supervised learning methods as an alternative to threshold-based methods which may provide more robust segmentation results. A major drawback of supervised learning is that it requires an annotated dataset. We present the data curation process and the deep learning approaches we tested.
Figure 5: Example of failure for threshold-based segmentation. Due to the two types of soils (inside and the tank), the distribution of color indexes has three modes making the choice of a threshold difficult. The middle picture shows the result of segmentation based on 2 different thresholds indicated on the color index distribution (left).

4.1. Building the dataset

For training models on a semantic segmentation task, a dataset should be annotated so that each pixel of the images can be attributed a label. We consider the case of two labels (plants and soil). Human annotators were asked to draw a polygon around each cultivated plant in a given image. The LabelMe\(^8\) software tool illustrated in Figure 6 was used for that purpose, running on a local server. The annotation process was considered long and painful by annotators. It took 20 minutes per image and we collected 15 annotated images.

We also considered automating the process using a generic object detector trained on the MSCOCO dataset. The output of a FasterRCNN network\(^9\), for example, are object proposals as bounding boxes for the various classes it was trained for. All these proposals were gathered and the constraint that the boxes should not overlap was added by taking the union of overlapping boxes. The results were

---


Surprisingly good although some plant were not detected. Another problem is that some plants were decomposed in two boxes or more when some leaves appear isolated (see Figure 6).

In both human or machine annotation, the output was not readily usable for training and a refinement strategy was necessary. For humans, the polygons were only coarsely following the contours of the plants. For machines, the annotations consisted of boxes far from a pixel-wise labeling. Using active contours\(^\text{10}\), we were able to obtain high resolution labeling of the images.

**Figure 6:** (Right) Web interface of the human annotation tool. (Left) Example of plant detection using a generic object detection network.

From the small set of clean labels we obtained, we trained a support vector machine to discriminate between plant and non-plant pixels. Inference was performed on a large dataset of images collected by the robot (273 images). Around 30% of the images resulted in a bad segmentation mask. Those were manually discarded, leaving a set of 193 images having a good segmentation mask.

5. Deep learning approach to semantic segmentation

More recently, we applied neural networks with deep architectures to the semantic segmentation problem and we tested some of these architectures on our dataset.

Convolutional neural networks initially designed for classification can adapted into semantic segmentation architectures, for example by transforming the last fully connected layers into convolutional layers. The SegNet model\textsuperscript{11} builds an encoder-decoder architecture based on the VGG16 classification network. In the encoder part of the network, convolutional layers and max-pool layers are interleaved reducing the resolution of feature maps along the way. In the decoding part, feature maps are upsampled by combining convolutions with unpooling. Each max-pool layer of the encoder has a corresponding layer in the decoder. At the unpooling step, the values of the previous layer are copied to units of indices selected as the maximum of the corresponding max-pool layers. The network is trained using a pixel-wise cross-entropy and is evaluated using the mean intersection over union. We used the 193 annotated images to train such a segnet instance and the output of the network on a test image is illustrated in Figure 7. We see that a large amount of pixels are wrongly classified as plants.

As we are concerned about our algorithm to be embedded on a mobile platform, we also looked at architectures which are less demanding in terms of computation and memory. Such a model based on ResNet blocks and factorized convolution kernels was developed in the Flourish european project\textsuperscript{12}. An annotated dataset for semantic segmentation of sugar beet crops was also made available during this project. We tested, on our data, the inference of the Bonnet architecture\textsuperscript{13} trained on the Flourish dataset (see Fig. 7). Similarly to the result of SegNet, the output shows also a low false negative rate with plants being detected as weeds.

The results obtained here could be improved in two ways. For the SegNet architectures trained on our annotated dataset, we expect better results by increasing the size of the dataset. We could also train the Bonnet architecture on our dataset. Another possibility is to use a network trained on the sugar beet dataset and to fine tune the model on our limited dataset to adapt the domain of segmentation of the network. This approach would not require additional data annotation and it seems promising given the result of the Bonnet inference without fine tuning.

\textsuperscript{12} \url{http://flourish-project.eu/}
6. Improving segmentation using the 3D structure

Although segmentation in 2D is already being used in our weeding application, it suffers from significant limitations. First, the contours of the plants obtained from the 2D images are not accurate since the perspective projection of plants depends on their height. Second, in the 2D framework, we suppose the soil to be flat and we don’t vary the altitude in the tool path whereas the height of the soil usually varies.

As a first step toward solving those issues, we present the 3D reconstruction of the workspace and an improved segmentation algorithm based on 3D data.

6.1. 3D reconstruction of the workspace

We use a structure-from-motion algorithm pipeline\textsuperscript{14} to reconstruct the 3D scene based on a sequence of 2D RGB images. During this process, the camera position of each picture is estimated by matching features in each pair of overlapping images. A dense point cloud is then obtained using stereo matching techniques. In the example shown in Figure 8, we used a sequence of 484 images around the workspace resulting in more than 12 millions points reconstructed.

6.2. Detection of plants and projection of the 3D point cloud

The same color index described in section 1 is used to detect the plants. The points classified as soil are used to estimate the plane on which plants will be projected. The first and second axis of a principal components analysis give the vectors defining the plane best approximating the soil. The point cloud of

---

\textsuperscript{14} Schonberger, Johannes Lutz and Frahm, Jan-Michael, Structure-from-Motion Revisited, Conference on Computer Vision and Pattern Recognition (CVPR) 2016
the segmented plants is then projected on this plane to give the orthographic projection of plants on the soil.

A mask is extracted from this projection and it is compared with the mask obtained by the perspective projection of the camera. Figure 9 shows a comparison of the contours extracted from the camera image with the contours extracted after projecting the 3D point cloud. It shows the differences between the perspective and orthographic projections are larger as we move away from the optical center of the camera. The variations in altitude of the soil are also represented on Figure 9. If the tool path doesn’t take these variations into account, the tool will sometimes go too deep in the soil or remain in the air above the soil.

7. Conclusion

The current pipeline, with threshold-based segmentation of 2D images, was tested successfully in our experiments. We also saw some limitations of this method and we propose two possible ways to overcome it: using training based methods and using 3D information. For the training based method, we will test whether domain adaptation of a network trained on external dataset is feasible or whether a large annotated dataset should be constituted. For the integration of 3D data, we relied on structure-from-motion techniques applied to 2D images. This requires heavy computation that was done in an offline manner. Future work will be dedicated to test a similar processing online with data from a 3D sensor.
Figure 9: (Left) Contours extracted from the 3D projection of the point cloud are shown in red on the camera picture. (Right) Portion of the point clouds classified as soil, the colors represent the variations in height.