Satellite Images-based Obstacle Recognition and Trajectory Generation for Agricultural Vehicles

Regular Paper

Mehmet Bodur and Moein Mehrolhassani

1 Eastern Mediterranean University, Famagusta, North Cyprus, Turkey
*Corresponding author(s) E-mail: moein_mehr@yahoo.com

Received 29 June 2015; Accepted 28 November 2015

DOI: 10.5772/62069

© 2015 Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

In this study, a method for the generation of tracking trajectory points, detection and positioning of obstacles in agricultural fields have been presented. Our principal contribution is to produce traceable GPS trajectories for agricultural vehicles to be utilized by path planning algorithms, rather than a new path planning algorithm. The proposed system works with minimal initialization requirements, specifically, a single geographical coordinate entry of an agricultural field. The automation of agricultural plantation requires many aspects to be addressed, many of which have been covered in previous studies. Depending on the type of crop, different agricultural vehicles may be used in the field. However, regardless of their application, they all follow a specified trajectory in the field. This study takes advantage of satellite images for the detection and positioning of obstacles, and the generation of GPS trajectories in the agricultural realm. A set of image processing techniques is applied in Matlab for detection and positioning.

Keywords Autonomous Vehicles, Labelling, Obstacle Detection, Image Processing, Field Robotics

1. Introduction

Rapid developments in computational intelligence, which are parallel to developments in digital science and technology, launched a new age for the automation of traditionally manual industries including agricultural production. Higher quality and volume in agricultural production requires a higher efficiency in ploughing, seeding, harvesting and similar traditionally manual tasks [40]. Various types of automated agricultural vehicles may be needed and implemented, requiring the tracking of a specific trajectory in the field.

Studies on the low-level control of autonomous vehicles propose several methods to track with a sufficiently low lateral error [29, 24]. A higher level of agricultural automation requires the automatic preparation of the reference paths to be followed by autonomous agricultural vehicles. Path planning algorithms, which are based on splitting a field into trapezoidal subfields and merging them into larger blocks to solve routing by a straight line, is covered by [39]. Path planning algorithms based on level set methods are analysed in [5] to improve re-computations in the case of environmental changes. A dynamic path search algorithm to guide tractors on turns at the end of fields is
considered in [43]. Various optimization criteria, such as B-patterns for optimal route planning with ant colony optimization, have been introduced in [1]. A probabilistically robust path planning algorithm using rapidly exploring random trees is introduced in [27] for the efficient identification and execution of paths in real-time. A complete coverage path-planning algorithm is proposed to generate dynamic paths, which avoid obstacles for a cleaning robot using neural networks [37]. A four-layer path-planning algorithm is developed in [34] to ensure the optimal real-time planning of multiple semi-autonomous vehicles. Cost function minimization and heuristic associated with a means of a greedy algorithm is used to determine the best Hamiltonian solution in field coverage path planning [10]. The DFS (Depth-First Search), with the use of sub-regions to avoid the overabundance of turns in narrow areas, is used by [13]. An algorithm to determine the turning areas, whilst minimizing overlapping, is introduced in [32]. A new approach to determine short- and long-term dynamic accuracy is proposed in [40]. Some methods for using GPS to control vehicles are proposed in [33, 36]. In addition, some approaches are introduced to detect objects physically [20, 4] and by imaging [38, 3, 9].

Our principal contribution is to propose an automated method for producing trajectories and the detection of obstacles, particularly trees, for an autonomous agricultural planting system, rather than a new path planning algorithm. In the remainder of this paper, Section 2 introduces the preliminaries for the satellite images and image processing for detecting and positioning obstacles in an agricultural field. Section 3 describes the proposed method in detail, and the results of this study are presented in Section 4. Section 5 contains a discussion of the proposed method and Section 6 presents our conclusions.

2. Preliminaries

In this study, a set of predetermined methods, from image processing techniques to GPS positioning systems, are used to achieve the stated goals. These tools, which are adapted to meet our needs, are introduced in this section.

2.1 Source of Satellite Images

Access to live satellite providers and images is limited due to national security restrictions. Google Maps is utilized in this study as the primary search engine. Images from Google Maps are assumed to be live and accurate for evaluation purposes, which is essential to demonstrate the expandability of the proposed method to use live satellite image providers. In this study, it is assumed that mappings between actual GPS coordinates of the earth and pixels of the images taken by Google Maps were done precisely by Google. A similar API-V3 Key, which is shown in Figure 1 and is specified by Google, is used to locate and import satellite images using the GPS coordinates of the field [12].

2.2 Positioning System

Google Maps uses GPS and Mercator projection to represent the map in tiles of 256 x 256 pixels, which gives us the whole earth in a tile at zoom level zero, as shown in Figure 2 [12]. Using a unique x, y pair, each tile can be accessed for positioning. In this system, (0,0) refers to the northwest of the map, based on Mercator projection.

2.3 Image Processing Techniques

The geographic satellite images of the target fields are processed using noise reduction, edge detection, colour reduction and median filtering methods to prepare the image for the determination of the boundaries and detection of obstacles.

2.3.1 Filtering and Noise Reduction

Median filtering is a nonlinear operation that is used to eliminate sharp singular marks, which are of no consequence due to their small size [8]. This filtering is preferred to others due to its potential to preserve edges, while suppressing noise.

2.3.2 Edge Detection Techniques

Edge detection is essential to find boundaries of the desired field and determine the specifications of obstacles like dimensions and location. It also helps to find the field’s attributes such as length and position. Several edge detection techniques, including Prewitt, Sobel, Roberts and Canny, are utilized [25, 11]. After noise reduction, Prewitt, Sobel and Roberts had more acceptable results, based on the parameters used in this study and presented in Section 3.2. As shown in Figure 3, among these popular methods, Sobel - with no thinning - seems to perform the best in the detection and presentation of edges in an agricultural environment.
3. System Description

The proposed method mainly works on top-view satellite images, and is processed to detect the boundaries of the field and the obstacles, as described in [23]. Although different kinds of obstacles, such as stones and trees, may exist in an agricultural field, this study focuses on detecting trees to emphasize the validity of this method. The system contains four main stages, as shown in Figure 4, consisting of: 1) initialization, to locate and import the image; 2) field extraction, in which the target field is extracted; 3) obstacle detection, to detect, position and store obstacles; and 4) trajectory generation, which generates tracking trajectory points.

3.1 Field Extraction and Sectioning

Users entering any GPS coordinates of the agricultural field do the initialization of the system to target the desired field. This study takes advantage of thresholding methods, such as Otsu’s method, to perform segmentation on the image for extraction of the target field and detection of obstacles [44]. The user determines the highest possible zoom level (Δz) that fits the field into the frame. The obtained image is converted to gray-scale for further processing by retaining the luminance, while eliminating the hue and saturation information. Conversion to gray-scale is obtained using the weighted sum of R, G, and B components of the imported colour image (0.2989 * R + 0.5870 * G + 0.1140 * B) [41].

On the gray image, a median filtering with a 3-by-3-neighbourhood matrix is applied to reduce typical “salt and pepper” noises [8]. The target field is extracted from the imported image using segmentation techniques and sectioning [30, 6]. The proposed method also allows the user to tune the filtering parameters of the automated process interactively, which gives the user an opportunity to select the target field manually if needed. Then, a binary form of the image is constructed, as in Figure 5b. The detected regions (Ri) are numbered, based on the number of connected pixels (ni), as in Figure 5c. The centre of the image is mapped to the GPS coordinate of the field, and the largest region, which overlaps it, is considered to be the target field [42]. The imported image is cropped using the bounding box properties of the field to only show the desired field, as in Figure 5d and 5e.

3.2 Obstacle Detection and Positioning

In this section, a method for detecting obstacles in an agricultural field is presented [23]. The undesired regions inside the field, which are too small to be deemed as obstacles, are removed using morphological image reconstruction by filling them [22, 16]. Top-view of trees in satellite images usually appear as a circle or ellipse, and are detected using Circular Hough Transform (CHT) [31]. The radius of trees has various ranges in different zoom levels. In this study, the CHT parameters are set to: object-polarity = ‘bright’, computation method = PhaseCode, sensitivity-factor = 0.9 and, finally, edge-gradient-threshold = 0.27.

Past experience shows that at Δz = 18, which this experiment is based on, trees have a radius range from 4-18 pixels, which changes in various zoom levels. The proposed method is designed to work with other zoom levels too. However, due to limited space in this article, the results will not be presented. Each detected circle to be accepted as a new tree entry is checked for duplicity. This is done by

Figure 3. Results of edge detection techniques. a) Sobel, b) Prewitt, c) Roberts and d) Canny

Figure 4. Architecture of the proposed system containing four stages: initialization, field extraction, obstacle detection and tracking trajectory point generation
mapping the centre point of the object to a pixel in the image and searching the list to find duplicates. In the case of no duplication, the tree is stored with its centre pixel and radius (in pixels) to determine its diameter.

Captured satellite images are not set with GPS information. Therefore, we have proposed a positioning method to convert pixels to GPS coordinates. The system works on a geographic coordinate system, which is based on latitude and longitude. Starting with the earth’s circumference at equator $a = 40075017$ metres for $\theta_e = 360^\circ$, the number of pixels in a tile $p_x = 256$ and $p_y = z \times 2^3$ the number of pixels in zoom level ($\Delta z$) and the length of one degree of Longitude Per Pixel (LDP) is determined by dividing $\theta_e$ over $p_z$. Additionally, by dividing $a$ over $p_z$, the number of metres per pixel at the equator is calculated, which is $0.5971645$ metres at $\Delta z = 18$. While moving from the equator to the poles, the distance between longitudes gradually approaches zero, which means that the distance of one degree at the equator has its maximum value of $111319.49167$ metres, and decreases as it approaches the poles, which was calculated in our previous study [23].

In addition, while moving towards the poles, the number of GPS points per pixel differs too. Therefore, to get an accurate position, another mapping is required. Multiplying latitude at each degree by the number of GPS points per pixel results in establishing an accurate position. The centre pixel of the image is mapped to the GPS point, which is provided by the user, as in Figure 6a. However, during field extraction, while cropping the image, the Original Centre Pixel (OCP) might be displaced, as in Figure 6b. In order to get an exact location, the offset needs to be calculated. From Figure 6b, the displacement of the centre pixel is obvious. To calculate this offset, Bounding-box’s Start Point (BSP) and Bounding-box’s End Point (BEP) (the dashed lines in Figure 6a) are used.

In this case, the values are (BSP ($X_o, Y_o$)) = (163.19, 173.11). The cropped centre pixel’s new position and offset are calculated by subtracting OCP’s coordinates from CCP’s. The only GPS coordinates of the field provided by the user in latitude ($r_{\text{lat}}$) and longitude ($r_{\text{long}}$) are mapped to the centre pixel of the image, which all further positioning procedures depend on, hence, maintaining its exact location is essential. Having the offset calculated, the system can calculate each detected tree’s exact coordinates in latitude and longitude pairs, using their centre pixel coordinate ($t_{\text{lat}}$, $t_{\text{long}}$) using Eq. 1 and Eq. 2. After validation, the object is stored with its properties, such as index, position, pixel coordinates, radius, latitude and longitude, and is graphically annotated with a red circle [19, 14].

$$t_{\text{lat}} = r_{\text{lat}} + \left( (\text{CCP}_y - ty_t) \times \cos \left( \frac{\text{CCP}_x \times \frac{\theta_e}{180}}{\text{Pixel}} \right) \right)$$  \hspace{1cm} (1)

$$t_{\text{long}} = r_{\text{long}} - \left( (\text{CCP}_y - ty_t) \times \frac{\theta_e}{\text{Pixel}} \right)$$  \hspace{1cm} (2)

3.3 Trajectory Points Generation and Positioning

The system provides a sequence of GPS trajectory points for agricultural vehicles to follow [35]. Having obtained all
of the required data that are related to the field and obstacles therein, the sequence of tracking trajectory points in the form of a path is ready to be generated. The aim is merely to provide trajectories rather than path planning. To test the integrity of the proposed approach, the generated trajectory points are integrated into a simple path planning technique, which finds the longest edge and provides parallel lines according to that edge.

In the binary image, it is assumed that the pixels with value “one” (white colour) are referring to the main field, and the trajectory points should be within these areas. The detected trees are re-plotted into the field using their centre point coordinate and radius in pixels as black regions with “zero” value after the field is filled with “one” values. This process results in a clearer, smoother and more precise image, containing both the field and the trees. Applying Standard Hough Transform (SHT) algorithm on the binary image results in finding the longest edge of the field. In the algorithm, the distance between the origin and the line $(p_i)$ is set to 0.5 and the angle from the origin to the closest point $(\theta_i)$ is set to $-90 \leq \theta_i < 90$, respectively. During the search, each line’s length is calculated using its first point coordinate $(p1)$, last point coordinate $(p2)$ and orientation angle $\theta_h$ (Eq. 3).

$$l_{line} = \sqrt{(p2_x - p1_x)^2 + (p2_y - p1_y)^2}$$  \hspace{1cm} (3)

A line with maximum $l_{line}$ value is stored as the longest edge $(ED_{max})$ along its $\theta_h$. The slope of the longest line $(m_{ED_{max}})$ is calculated, as follows:

$$m_{ED_{max}} = \frac{p2_y - p1_y}{p2_x - p1_x}$$  \hspace{1cm} (4)

The $ED_{max}$ is highlighted with a blue bold line, as shown in Figure 7a. Desired paths must be parallel to the $ED_{max}$ with equal distance from each other. The distance between paths $(d_{pix})$ in metres is converted to pixels using Eq. 5.

$$d_{pix} = \frac{256 \times d_{max} \times 2^{\theta_y}}{\theta_y}$$  \hspace{1cm} (5)

The desired paths are to be parallel to each other with constant $d_{pix}$ as entered by the user. Therefore, temporary points (red dots) are plotted as origin points to draw paths, as in Figure 7b. Any deviation of origin points will result in lines with unequal distance $d_{pix}$ from each other. Midpoint $(p_{mid_x,y})$ of the $ED_{max}$ is taken as the start point to draw the perpendicular lines.

$$p_{mid_x,y} = \left(\frac{p1_x + p2_x}{2}, \frac{p1_y + p2_y}{2}\right)$$  \hspace{1cm} (6)

$$OP_x = p_{mid_x} + \left(\Delta inc \times \left(\sin m_{ED_{max}}\right)\right)$$  \hspace{1cm} (7)

$$OP_y = p_{mid_y} + \left(\Delta inc \times \left(\cos m_{ED_{max}}\right)\right)$$  \hspace{1cm} (8)

The slope of the perpendicular line is $m_{perp} = \tan (m_{ED_{max}})$. Each origin point $(OP_x,y)$ is placed one by one with a constant increment $(\Delta_{inc})$ to generate points with equal distance from each other using Eq. 7 and Eq. 8. The system generates tracking trajectories using the longest edge, perpendicular line and origin points. A sequence of trajectory points in a straight line is generated by replacing $m$ with $m_{perp}$ in linear equation $y = mx + b$ Eq. 9 and Eq. 10

$$PL_x = 1 to image width$$  \hspace{1cm} (9)

$$PL_y = \left( m_{perp} \times (PL_x - OP_x) + OP_y \right)$$  \hspace{1cm} (10)

These pixels represent the points in which tracks parallel to the longest edge should pass through. Any intersection between the trajectory points and detected obstacles in the field needs to be determined to avoid any collisions. The system maps each generated point’s pixel coordinates $(x, y)$ to the image and compares their value and, if it is overlapped with a pixel with a “zero” value, the point is discarded. This is due to the fact that either the generated point is placed on an obstacle or it is outside of the field’s boundary, as both are presented with “zero” value. After passing all of the conditions, the trajectory points are plotted sequentially on the field as red dots in straight lines, as in Figure 7a.

```
Figure 7. Tracking trajectory point generation and representation in a) general, b) detail
```

Each trajectory point is converted from pixel to GPS coordinates to be used by a GPS tracking enabled agricultural vehicle [23]. As indicated, moving between the poles
and equator results in different distances that each pixel can cover. Therefore, the exact length of a degree of longitude per pixel (LDPP) must be calculated [23]. Pixel-to-GPS conversion results are stored in a 2D-array (m × n)Table 1. Columns 1 to 4 contain information related to the first track, columns 5 to 8 contain information for the second track and so on. The first two are pixel coordinates (\( \mathcal{P}_X, \mathcal{P}_Y \)) and the next columns are GPS coordinates (\( \mathcal{P}_{\text{Lat}}, \mathcal{P}_{\text{Long}} \)) of the trajectory point.

| Trajectory Points | X     | Y     | Lat. | Long. |
|------------------|-------|-------|------|-------|
| 23               | 63    | 12.391... | 35.0755... | 33.5308... |
| 24               | 64    | 12.836... | 35.0755... | 33.5308... |
| 25               | 65    | 13.281... | 35.0755... | 33.5308... |
| 26               | 66    | 13.725... | 35.0755... | 33.5309... |
| 27               | 67    | 14.170... | 35.0755... | 33.5309... |
| 28               | 77    | 18.619... | 35.0755... | 33.5309... |
| 29               | 78    | 19.064... | 35.0755... | 33.5309... |
| 30               | 79    | 19.509... | 35.0755... | 33.5309... |

Table 1. Format of the path’s trajectory points in an array

4. Results

In order to illustrate the proposed method, a number of examples are presented. The system is developed and tested on MATLAB technical programming language (The MathWorks Inc., Natwick, MA) platform. A computer with a CPU of 2.8 GHz (Intel Core i7) and 8 GB of RAM running on a Mac OS X Yosemite (10.10.3) is used to run and test the system. All of the satellite images are obtained from Google Maps Image API with resolution of 640x640 pixels, as provided by Static Map API. The algorithm was tested on seven different fields with various shapes, sizes and number of obstacles to validate the system. The zoom level in all of the tests is set to 18 for comparison purposes. In addition, the radius range for the detection of trees regarding \( \Delta \alpha = 18 \) is set to a minimum of \( \mathcal{R}_{\text{min}} \approx 4 \) and a maximum of \( \mathcal{R}_{\text{max}} \approx 18 \) in pixels. However, this is subject to change depending on zoom level. After an extraction of each field, the original image is cropped to only contain the desired field, resulting in an image with different dimensions, as in Figure 8. Furthermore, the tree detection outputs of these fields are graphically presented with red circles in the images.

Details of the detection section for these seven fields are shown in Table 2. The columns from left to right are the total number of trees, the total number of tree detections, correct detections, incorrect detections, the accuracy of correct detections, the percentage of incorrect detection and the complexity of the field. Complexity is defined based on the number of obstacles, their shape, size, fields’ dimensions and smoothness of edges. The distance between the desired paths are set to be \( d_m = 2 \) m in all of the fields for comparison purposes. Results of the path finding algorithm based on SHT are shown in Figure 9.

5. Discussion

According to the experimental results, the system is capable of detecting obstacles with significant precision, while using the free Google Maps API, which provides a maximum 640x640 pixels image and forced us to use low zoom levels. The system’s accuracy is correlated with zoom level, such that a higher zoom level results in more precise detections. Using Google Business API with an image of 2048 x 2048 pixels, the same experiment could be conducted with zoom level 20, resulting in an image with four times more clarity, as in Figure 10. In turn, this would increase the precision of detection. As previously presented in Table 2, the results indicate that, while the number of trees and complexity increases, the accuracy remains significant.

Figure 8. Test fields after crop and obstacle detection

Figure 9. Results of generating sequence of trajectory points to form the desired path

Overall, the accuracy is above 80%, except for fields 1 and 7. The low accuracy in field 1 is due to the fact that one of the two trees in the field is covered by Google’s watermark.
on the image, making the system unable to detect the tree. Additionally, in field 7, since many of the trees were together and formed, non-circular shapes resulted in the failure of detection. Table 2 shows that the average accuracy in the detection of trees and average error are more than 87% and less than 3% respectively, which represents a significant result.

Assuming that Google provides precise data, the proposed system proves to be successful in positioning. Coordinates of an obstacle detected and positioned by the proposed method is searched by Google Maps to validate its accuracy as an example. Figure 11 shows that the GPS coordinate points exactly to the centre of the detected tree.

The proposed method is also accurate in calculating the detected obstacles’ sizes, although they may appear in different sizes in the agricultural fields. As an example, the detected trees’ diameter using the proposed method (Figure 12a) is 10.14 m (20.74 pixels in the image), from Google Maps is 10 m (Figure 12b) and its actual diameter, as recently measured, was 11.1 m (Figure 12c). A comparison between the detected trees’ sizes calculated by the proposed method and Google Maps (14 cm difference) indicates a very low error rate in measuring obstacle sizes. However, the big difference between the actual size of the obstacle and the size provided by Google Maps indicates the insufficiency of Google Maps’ accuracy in representing data.

The path planning strategy used in this study generates paths that are parallel to each other. A linear equation of the base line with different y-intercept but constant increment is considered to guarantee the parallelism and straightness. Both tracks in the pixel and GPS coordinates are compared to validate the accuracy of conversion. From the line equations in the charts presented in Figure 13, it can be concluded that both lines in the pixel and GPS coordinates are straight lines and this is due to the fact that all of the exponents are not higher than one.

Furthermore, the coefficient of determination ($R^2$) of the points with the value of 1 confirms the complete straightness of the both lines. The coefficient of determination of the trajectory points in both the pixel and GPS coordinates for all fields in this study are presented in Figure 14a. The results indicate that the points are less that 0.005% variation from the straight line, which shows that the conversion is significant.

In addition, it is necessary for these points to be equidistant from each other, which ensures that tracking devices would not lose or stray from the path. A low standard deviation result indicates the accuracy in generation and conversion.
of tracking trajectory points, as shown in Figure 14b. To investigate the computational times, the same experiments are performed with various distances \((d_m=2m, d_m=4m, d_m=6m)\) and the results are shown in Figure 15, which are in the range of 2.04s to 25.95s for \(d_m=2m\), 1.35s to 14.47s for \(d_m=4m\), and 1.15s to 10.69s for \(d_m=6m\). The results indicate that the computational time increases almost twofold by decreasing the distance between paths to half.

Table 3 presents the processing time for all of the fields. From the table, it can be observed that the highest rates of processing time belong to fields 3 and 4, as they are almost twice the size of fields 1, 2, 5 and 6, which requires more trajectory points to cover the field.

![Figure 14](image-url)

**Figure 14.** a): Coefficient determination of x,y and GPS coordinates. b) Standard deviation of the distance between tracking trajectory points in x, y and GPS coordinates.

![Figure 15](image-url)

**Figure 15.** Computational time for 2, 4, and 6 m path distance in seven fields

Although field number 7 is almost the same size as fields 3 and 4, the results from Table 2 indicate that the computational time and generated trajectory points are half as many. Table 1 shows that the system has detected one obstacle in field number 3, and three in field number 4, while, for field 7, there are 194 detected obstacles. Having more obstacles in the field results in less space for planting, which causes the system to eliminate those areas from the path. This results in a decrease in generation of tracking trajectory points and computational time.

6. Conclusion

In this study, a method for the generation and positioning of tracking GPS trajectory points for agricultural autonomous systems has been presented. The proposed method begins with a single GPS coordinate of an agricultural field, and detects obstacles automatically. The acquired global position and dimensions of detected obstacles can be used for agricultural vehicles to avoid any collisions. Low computational time in the range of 1.15 – 25.95 seconds, along with simplicity of use and accuracy in positioning and generating of tracking trajectory points, makes the proposed method feasible to be integrated into any path planning algorithm or method for automation of agricultural vehicles. The generated trajectory points are graphically represented in the image to be presented to the user. This method is tested with a simple path planning technique to validate the generation of trajectory points. According to the experimental results, the proposed method provides significant means to generate trajectory points and position obstacles. In the near future, this method can be extended to cover more types of obstacles, along with the development of other subsystems of agricultural automation such as interactive user interfacing for obstacle verification.

7. Acknowledgements

The authors wish to thank Anas MAHDI for his contribution to this study in the statistical analysis.

8. References

[1] Bakhtiari A, Navid H, Mehri J, Bochti D: Optimal Route Planning of Agricultural Field Operations Using Ant Colony Optimization. Agricultural Engineering International: CIGR Journal. 2011; 13, No.4.

[2] Anita, Liang, Gauri, Salokhe, Margherita, Sini, Johannes, Keizer: Towards an Infrastructure for Semantic Applications: Methodologies for Semantic Integration of Heterogeneous Resources. Cataloging & Classification Quarterly. 2006:43:161-189.
[3] Anuj, Mohan, Constantin, Papageorgiou, Tomaso, Poggio: Example-Based Object Detection in Images by Components. IEEE Transactions on Pattern Analysis And Machine Intelligence, 2001;23(4): 349-361. DOI: 10.1109/34.917571.

[4] Antonio, Sgorbissa, Renato, Zaccaria: Integrated Obstacle Avoidance and Path Following Through a Feedback Control Law. J Intell Robot Syst. 2013;72:409-428. DOI 10.1007/s10846-012-9806-2.

[5] Bin X, Daniel J-Stilwell, Andrew J-Kurdila: Fast Path Re-planning Based on Fast Marching and Level Sets. J Intell Robot Syst. 2013;71:303-317. DOI: 10.1007/s10846-012-9794-2.

[6] Chen W, Tao-Ruan W, Ian J-P: Automatic Reconstruction Of 3D Environment Using Real Terrain Data and Satellite Images. Intelligent Automation and Soft Computing. 2012;18(1):49-63. DOI: 10.1080/10798587.2012.10643226.

[7] Dragos B, Ion N, Mihai D: Bridging the Semantic Gap for Satellite Image Annotation and Automatic Mapping Applications. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 2011;4(1): 193 - 204. DOI: 10.1109/JSTARS. 2010.2081349.

[8] Maheswari D, Radha V: Noise Removal in Compound Image Using Median Filter. IJCSSE International Journal on Computer Science and Engineering. 2010; 2(4):1359-1362.

[9] Erik C, Noé Ortega-S, Daniel Z, Marco Pérez-Cisneros: Circle Detection by Harmony Search Optimization. J Intell Robot Syst. 2012;66:359-376. DOI 10.1007/s10846-011-9613-3.

[10] Fabre S, Philippe S, Michel T, Lionel C: Framework Path Planning for Field Coverage with Minimum Overlapping. International Conference on Emerging Technologies and Factory Automation. Proceedings. IEEE;2001, P. 691-694.

[11] Shrivakshan T, Chandrasekar C: A Comparison of Various Edge Detection Techniques Used in Image Processing. IJCSI International Journal of Computer Science Issues. 2012;9(5-1), ISSN (Online): 1694-0814.

[12] Google Map API. Available from: https://developers.google.com/maps/documentation. Accessed on 23 Dec 2013.

[13] Guoyu Z, Peng Z, Junfei Q: Path Planning Algorithm Based on Sub-Region for Agricultural Robot. 2nd International Asia Conference on Informatics in Control, Automation and Robotics; 6-7 March 2010; Wuhan; 2012. P. 197-200.

[14] Hei-Chia W, Tian-Hsiang H, Chia-Tzung L: Automatic Summarization Based on Automatically Induced Ontology. Intelligent Automation and Soft Computing. 2011;17(4):447-463. DOI: 10.1080/10798587.2011.10643160.

[15] Hameed I, Bochtis D, Sorensen C, Neremark M: Automated Generation of Guidance Lines for Operational Field Planning. Bio Systems Engineering, 2010;107;294-306. DOI: 10.1016/j.biolsyseng. 2010.09.001.

[16] Jian-Jone C, Chun-Rong S, W-Eric G, Jun-Lin L, De-Hui S: Object Segmentation of Database Images by Dual Multiscale Morphological Reconstructions and Retrieval Applications. IEEE Transactions on Image Processing. 2012;21(2):828-43. DOI: 10.1109/TIP.2011.2166588.

[17] Jacek K, Tomas V, Carine B, Joel F: SAWSDL: Semantic Annotations for WSDL and XML Schema. Internet Computing. IEEE. 2007;11(6):60-67. DOI: 10.1109/MIC.2007.134.

[18] Kevin M, Antonio T, Daniel E, William F: Object Detection and Localization using Local And Global Features. Toward Category-Level Object Recognition Lecture Notes in Computer Science Volume 4170; 2006. P. 382-400. DOI: 10.1007/11957959_20.

[19] Kuhandhama M, Michael N-H: Ontology Tools for Semantic Reconciliation in Distributed Heterogeneous Information Environment. Intelligent Automation and Soft Computing.2000. p. 1-8.

[20] Kise M, Noguchi N, Ishii K: Laser Scanner-Based Obstacle Detection System for Autonomous Tractor-Movement and Shape Detection Targeting at Agricultural Vehicle. Journal of the Japanese Society of Agricultural Machinery.2005;66(2):97-104.

[21] Kosmas P, Stephan B, Carsten S, Nikos S, Stamata D, Vassilis T, Siegfried H, Yannis A, Yiannis K, Steffen S: Knowledge Representation and Semantic Annotation of Multimedia Content. IEEE Proc.-Vis. Image Signal Process.2006;153(3):255-262. DOI: 10.1049/ip-vis:20050059.

[22] Luc, Vincent: Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms. IEEE Transactions on Image Processing. technical report 91-16, Harvard Robotics Laboratory, November 1991, IEEE Transactions on Image Processing, 1993. P. 176-201.

[23] Mehmet B, Moein M: Textual Annotation, Detection and Positioning of Obstacles in Agricultural Fields Using Satellite Images. ICSCCW-2013 Seventh International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control. Izmir, Turkey September 2-3, 2013.

[24] Mehmet B, Ehsan K, Hasan H: Double Look-ahead Reference Point Control for Autonomous Agricultural Vehicles. Biosystems Engineering,. 2012; 113(2):173-186. DOI: 10.1016/jbiosystemseng. 2012.07.005.

[25] Mamta J, Parvinder S: Performance Evaluation of Edge Detection Techniques for Images in Spatial Sensing. Proceedings. IEEE;2001. P. 691-694.
[26] Milan G, Milan M, Bojicb S, Djordje D: Short- and Long-term Dynamic Accuracies Determination of Satellite-based Positioning Devices Using a Special- ly Designed Testing Facility. Computers and Electronics in Agriculture. 2011;76(2):297-305. DOI: 10.1016/j.compag.2011.02.008.

[27] Mangal K, Ian P: A Probabilistically Robust Path Planning Algorithm for UAVs Using Rapidly-exploring Random Trees. J Intell Robot Syst. 2013;71:231–253. DOI: 10.1007/s10846-012-9776-4.

[28] Marie L, Henri M, Mihai D: Semantic Annotation of Satellite Images Using Latent Dirichlet Allocation, IEEE Geoscience and Remote Sensing Letters, 31 July 2009. TELECOM Telecom ParisTech, Paris. 2010. P 28-32.

[29] Norrmarka M, Grieppentrog Hb, Nielseneb J, Sogaaerde H: The Development and Assessment of the Accuracy of an Autonomous GPS-based System for Intra-row Mechanical Weed Control in Row Crops. IAgRI. Published by Elsevier Ltd. 2008; 101(4):396-410. DOI :10.1016/j.biosystemseng. 2008.09.007.

[30] Muhammad K, Abdul G, Adil M-S: Image Segmentation Using Multilevel Graph Cuts and Graph Development Using Fuzzy Rule-based System. IET Image Process., 2013;7(3):201-211. DOI: 10.1049/iet-ipr.2012.0082.

[31] Marcin S, Ignacy D: Circular Object Detection Using a Modified Hough Transform. Int. J. Appl. Math. Comput. Sci. 2008; 18(1):85-91. DOI: 10.2478/v10006-008-0008-9.

[32] Michel T, Philippe S, Helene F, Lionel C: Path Planning for Complete Coverage with Agricultural Machines. Field and Service Robotics, Springer-Verlag Berlin Heidelberg 2006. P 549-558. DOI: 10.1007/10991459_53.

[33] Robert D, David M-B: The Use of GPS for Vehicle Stability Control Systems. IEEE Transactions On Industrial Electronics. 05 April 2004. p. 270 – 277.

[34] Rahul K, Kevin W: Multi-Level Planning for Semi-autonomous Vehicles in Traffic Scenarios Based on Separation Maximization. J Intell Robot Syst. 2013; 72:559–590, DOI: 10.1007/s10846-013-9817-7.

[35] Roland L, Benoit T, Christophe C, Philippe M: High Accuracy Path Tracking for Vehicles in Presence of Sliding: Application to Farm Vehicle Automatic Guidance for Agricultural Tasks. Autonomous Robots. 2006;21(1):79-97. DOI: 10.1007/ s10514-006-7806-4.

[36] Roland L, Benoit T, Christophe C, Philippe M: Adaptive and Predictive Path Tracking Control for Off-road Mobile Robots. European Journal of Control. 2007;13(4):419-439. DOI: 10.3166/ejc. 13.419-439.

[37] Simon X, Yang C-L: A Neural Network Approach to Complete Coverage Path Planning. IEEE Transaction on Systems Man and Cybernetics Part B: Cybernetics. 2004. p. 718-724

[38] Thomas D, Daniel K, Hermann N: Discriminative Training for Object Recognition Using Image Patches. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), Feb 2005.

[39] Oksanen T, Visala A: Path Planning Algorithms for Agricultural Machines. Agricultural Engineering International: The CIGR Ejournal. Manuscript ATOE 07 009. IX. Feb 2007.

[40] Victor B, Luisa C, Yolanda H, Justo P: Planning for Agricultural Forage Harvesters and Trucks: Model, Heuristics, and Case Study. Networks & Spatial Economics - NETW SPAT ECON. 2010;10(3): 321-343. DOI: 10.1007/s11067-009-9120-0.

[41] Wain B, Nyein A: On the Automatic Detection System of Stop Line Violation for Myanmar Vehicles (Car). International Journal of Computer & Communication Engineering Research (IJCCER). 2013;1(4), e-ISSN: 2321-4198 p-ISSN: 2321-418X.

[42] Yu-Wing T, Jiaya J, Chi-Keung T: Soft Color Segmentation and Its Applications. 29 September 2007 IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007. p.1520-1537.

[43] Zhang Q, Qu H: A Dynamic Path Search Algorithm for Tractor Automatic Navigation. Transactions of the ASAE. 2004; 47(2). DOI: 10.13031/2013.16027.

[44] Nobuyuki O: A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics. 1979 p. 62-66.