

# Automatic vectorization of rectangular manmade objects: a case study applying OpenCV and GDAL on UAV imagery

Márton Pál <sup>a,b</sup> \*, Fanni Vörös <sup>a,b</sup>, Béla Kovács <sup>b</sup>

<sup>a</sup> ELTE Eötvös Loránd University, Doctoral School of Earth Sciences, [marchello@map.elte.hu](mailto:marchello@map.elte.hu), [vorosfanni@map.elte.hu](mailto:vorosfanni@map.elte.hu)

<sup>b</sup> ELTE Eötvös Loránd University, Faculty of Informatics, Institute of Cartography and Geoinformatics, [climbela@map.elte.hu](mailto:climbela@map.elte.hu)

\* Corresponding author

**Abstract:** UAV imagery has a big role in environmental mapping: various indices regarding plant health, soil condition or geological objects can be determined, or 3D models can be built for accurate measurements. Automatic vectorization of satellite images is widely applied nowadays for land coverage determination purposes. However, larger resolution UAV images are hard to process following this theory: too many details result in a long computing time. We propose a FOSS (free and open-source software) analytical solution for detecting and vectorizing quasi-rectangular shaped (mainly manmade) objects on relatively high-resolution images. Our sample area is the cemetery and its surroundings in Istenmezeje, Heves County, Hungary. The graves are good examples of regular, rectangular manmade objects. The traditional cadastral mapping of these sites means a large amount of digitizing work. We have used Python environment for conducting image analysis: delineating and vectorizing the grave outlines for the large-scale mapping of the cemetery. Open-source programming libraries were used during the process: OpenCV and GDAL/OGR. With these tools, we were able to digitize the graves automatically with systematic errors. Approximately 70-80 of 100 graves were correctly recognised (their number varies depending on the adjustable variables: the size and detailedness of the contours to be detected). Our approach is a relatively new methodology in large-scale cartography: computer vision tools have not been used widely for mapmaking purposes. The development of artificial intelligence and open-source tools connected to it may contribute to the broader dissemination of similar methodologies in cartography and GIS.

**Keywords:** OpenCV, GDAL, Python, image processing, UAV

## 1. Introduction

The application of UAVs (Unmanned Aerial Vehicles) in industrial, defence, agricultural and scientific sectors is growing dynamically in the past years. As more and more possibilities emerge, the number of drone pilots and imagery users increases (Major et al. 2016, Restás 2017).

The analysis and application of remotely sensed data in cartography have started to evolve in the past decades. More and more GIS workflows use airborne imagery for environmental mapping, agricultural analyses, engineering purposes, and nature conservation issues (e.g., Christiansen et al. 2017, Yang & Hawthorne 2021). As the traditional, analogue mapmaking process cannot keep pace with the ever-changing natural/manmade features and objects of the world, cartography started to develop automatic methods to make maps (Kovács et al. 2021).

### 1.1 State of the art

The analysis and processing of remotely sensed images in cartography is a well-developed area. Automatic analyses and vectorisation of these images are widely applied in e.g. landcover determination and its spatio-temporal examination (Naaouf & Elek 2020, Lausch et al. 2018). UAV imagery has also a big role – but the high resolution often sets limits. The general scale of this survey method is large or medium – depending on the size of the UAV,

flying height and the aim of the mapping. The long computing time demanded by the detailed image may be shortened by the resampling of images – but it results in unexpected and inaccurate conclusions and outcomes.

Eventually, cartography has started to apply automatic vectorisation aided by computer vision (CV) and artificial intelligence (AI) in the past years. An international workshop was organised in 2020 by the ELTE Eötvös Loránd University in Budapest on this topic. Publications about the vectorisation of old maps (Gede et al. 2020, le Riche 2020, Kratochvílová & Cajthaml 2020) and the recognition of various point and area features (Dusek 2021, Gede et al. 2021) on images have been issued previously. However, direct vectorisation methodologies have not been presented yet.

### 1.2 Aims and sample area

Our paper deals with the automatic recognition and vectorisation of quasi-rectangular objects on UAV imagery using OpenCV and GDAL (Geospatial Data Abstraction Library)/OGR (OpenGIS Simple Features Reference Implementation) in a Python environment. The editing of cadastral maps takes a long time because of the large amount of digitizing work. By modifying the images with the proposed algorithm, and by the detection of these regular forms, the time effort significantly reduces.

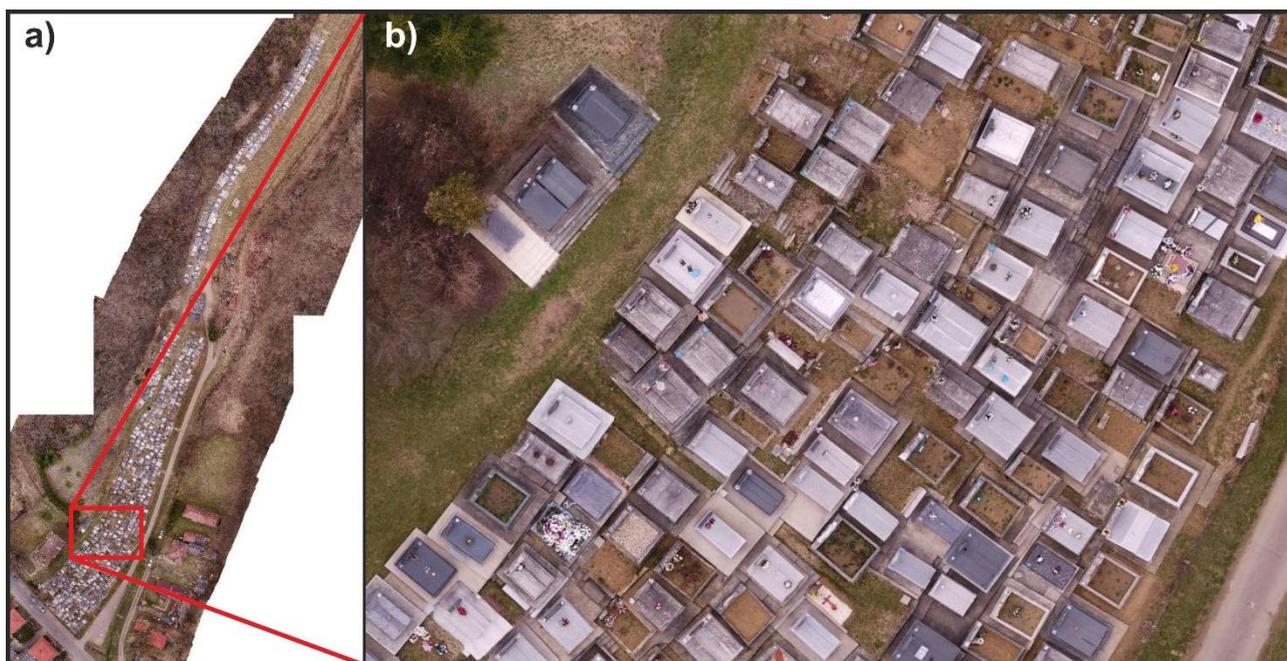


Figure 1. a) An overview of the cemetery's orthomosaic: it is a rather long and narrow area. b) The graves are unevenly distributed, and they are not rotated in the same direction.

Our sample area was the cemetery and its surroundings in Istenmezeje, Hungary (*Fig. 1*). The graves (covered with granite or other artificial stone frames) are good examples of regular, rectangular manmade objects. To make the situation more difficult, these objects are not regularly placed: they are unevenly distributed and rotated in different heights as the cemetery is placed on the side of a steep hill. The main aim of this study is to produce a vector dataset of the graves using the aforementioned GIS and computer vision tools. This database can be the base of a future cadastral map of the area that helps to parcel and administrate the sample area.

## 2. UAV flight mission

Two main factors affected our flight: the changing legal background and the environmental (weather and vegetation) conditions.

### 2.1 Legal background

Our DJI Matrice 210 RTK V2 (*Fig. 2*) quadcopter is a complicated tool: its legal use is not easy since the beginning of 2021 in Hungary. The application of two European Commission decrees (2019<sup>1</sup> and 2019<sup>2</sup>) is the task of each member country who can make more strict regulations to a certain extent.

When planning the mission, we had to make a checklist about the requirements that we needed to fulfil.

- Our sample area is within a nature conservation area. Because of this, we had to ask for permission from the local nature protection body. Flying without it is punishable.
- Every drone pilot who does not fly with toys (toys according to Hungarian law are under 120 g and without cameras) needs a license. Its level depends on the difficulty and risk of the

mission – in our case, minimum an A2 or a higher-level license was needed (it refers to the riskiest missions in the open category because of the built-in areas).

- As the sample area is within the borders of a built-in area, we needed to book airspace from the corresponding governmental organization. Before starting and after completing the flight, we needed to inform this authority via phone.
- The use of the 'mydroneSpace' (<https://mydroneSpace.hu/applikacio>) application is obligatory in any circumstances during the flight. It requires a continuous Internet connection to be accessible for the authorities in any dangerous situation (e.g., the unexpected flight of an ambulance helicopter over the area).

Additionally, from 2023, every UAV will be classified based on its weight and other capability attributions. Flying missions will be also restricted based on UAV categories.



Figure 2. Our DJI Matrice V210 RTK drone. The RTK GPS navigation promises cm-accurate flying and measurements.

## 2.2 Weather and vegetation conditions

The weather had to be taken into consideration when planning – and fortunately, we chose nearly a perfect day in March 2021. Although the weather was cold (but above 0°C), light conditions were ideal. The sky was overcast, which was slightly better than continuous sunlight because of the well-distinguishable colours. It was much better than the alteration of sunlight and clouds as the light conditions of the orthomosaic would have become unbalanced in this situation.

The vegetation is also important, as we did our survey with a simple RGB camera. These instruments cannot record surfaces under the vegetation (e.g., trees in contrary to LiDAR sensor to a certain extent). In the beginning of March, the trees did not start to bud, and local people did not put any wreaths or flowers on the graves yet due to nighttime freezes.

On the field, we have conducted a planned mission with 80% lateral and 80% frontal overlapping. After processing, we used the resulting orthomosaic for our automatic vectorisation purposes (*Fig. 1b*).

## 3. Methodology and resulting data

The task of delineating the graves was solved in a Python environment. As even custom libraries are easily installable, OpenCV and GDAL/OGR functions can also be applied.

The use of OpenCV may be a bit surprising. This programming function library mainly aims for real-time computer vision. Apart from motion tracking, augmented reality, gesture recognition and stereo vision, it has functions for segmentation, recognition and object detection too. It also contains a statistical machine learning library.

The GDAL and OGR is a much more GIS-friendly approach: these libraries contain operations for raster (GDAL) and vector (OGR) graphics (mainly with spatial reference).

These two large packages were used to build the following workflow.

### 3.1 Workflow

The orthomosaic was sliced up, as the work was done as a batch process due to the large size of the mosaic. In the first step, images (GeoTIFFs) were called by the program and passed to OpenCV.

A peculiarity of OpenCV is the default BGR colour format. Because of this, we need to convert our images to RGB first. After that, we converted it to grayscale as this simplifies the latter calculation and analysis process (*cv2.COLOR\_RGB2BGR*; *cv2.COLOR\_BGR2GRAY*).

After a simple noise filtering (*cv2.blur*), the grayscale image underwent a Canny edge detection (*cv2.Canny*). This edge-detected source file was used for the delineation of the bordering lines of the most important objects (*cv2.findContours*, *Fig. 3*). Initially, we presumed that this would be enough because of the threshold parameter of the edge detector, as smaller lines also

appeared (wreaths, benches or crucifixes). Although larger threshold values resulted in the reduction of the number of detected small objects, some graves also disappeared. Another problem was that this detector only found short line segments – their conversion to polygons is an inaccurate method due to the resulting small polygons.



Figure 3. Detected outlines marked by green lines. There are many unnecessary objects: flowers, wreaths, crucifixes and concrete pavements around some graves in the image.

These segments (above a certain length) are well-usable to determine the bounding rectangle of the graves (*cv2.approxPolyDP*). This step is parametrizable, as we can set the minimum searchable arc length and a minimum threshold for the detected rectangles' areas. In *Fig. 4ab*, some of the detected features can be seen. This setting was quite inaccurate: many of the important green border lines were filtered due to a too high arc length threshold value.



Fig. 4a. An excerpt of the sample area for detecting the borders of the graves and their bounding rectangles.

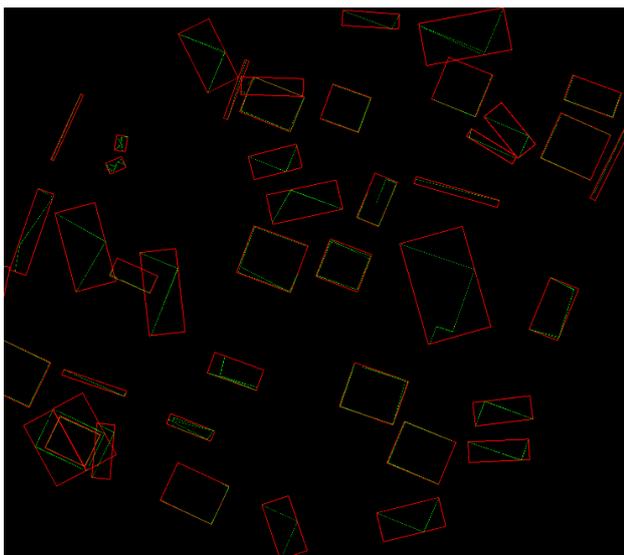


Fig. 4b. The result of the border and bounding rectangle with unsuccessful settings (many important border segments were filtered out due to a too high threshold value). The green lines are the contours of the graves, while the red polygons are the bounding rectangles.

OpenCV offers a possibility to watch the result of any steps (*cv2.imshow*). When the workflow is well-parametrized, and the bounding rectangle results are satisfying, we can save the images as GeoTIFFs - but this is the task of GDAL. After setting the file format and the projection, we get the raster band that contains the rectangles and write it out as an array (*gdal.GetDriverByName, Create, SetGeoTransform, SetProjection, GetRasterBand, WriteArray*).

As this GeoTIFF contains just the relevant information as pixel values, it can be converted to SHP (or other vector file type) format using GDAL/OGR (*ogr.GetDriverByName, CreateDataSource, outDataSource.CreateLayer, gdal.Polygonize*).

### 3.2 Resulting polygon dataset

By applying the above-described workflow, with correct and suitable parameters and threshold values, we were able to recognise 70-80 graves out of 100 (Fig. 5). It is important to note, that the results of any steps of the workflow can be written out or modified. E.g., if we are interested in the polygons of the grave contours, it can be also easily produced.

## 4. Discussion and conclusion

Our designated aim – detecting and delineating quasi-rectangular manmade objects (graves in this case study) – has been achieved by the workflow presented in this paper. The main steps and partial results were the following:

- Setting up a Python environment with OpenCV and GDAL;
- Converting the images to the appropriate colour model, detecting bordering lines and bounding rectangles with OpenCV;

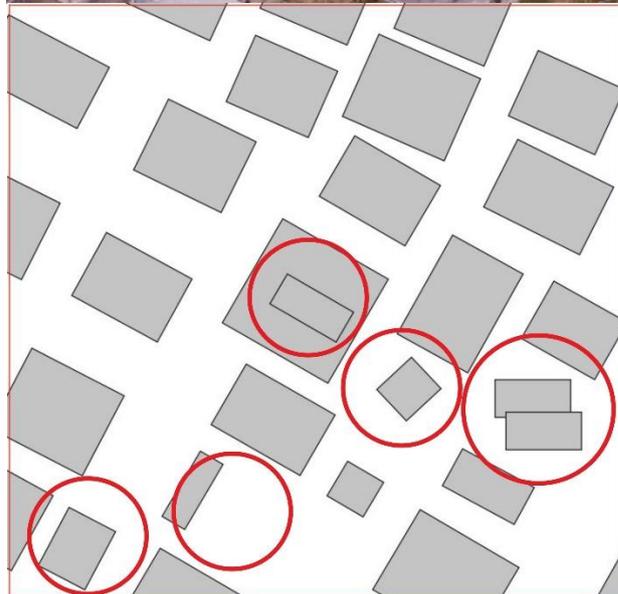


Figure 5. A detected polygon set of an excerpt of the cemetery. The accuracy is about 70-80%, the workflow proved to be useful when making the map of the area (the mistakes are in red circles).

- Passing these data to GDAL/OGR to provide spatial reference and converting the images to vector format.

Although a sketch map and a vector polygon dataset can be produced this way, the refinements and corrections are needed to be made manually. Additionally, the parametrisation is also problematic, as most sample areas need different settings depending on weather conditions, image resolution and most importantly the detectable objects.

However, this paper and methodology can contribute to the further dissemination and application of automatic vectorisation in cartography presenting a practical use of computer vision.

## 5. Acknowledgements

BK is supported by the NRD Fund of Hungary, Thematic Excellence Programme no. TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

MP and FV are supported by EFOP-3.6.3-VEKOP-16-2017-00001: Talent Management in Autonomous Vehicle Control Technologies – The Project is financed by the Hungarian Government and co-financed by the European Social Fund.

## 6. References

- Christiansen, M.P., Laursen, M.S., Jørgensen, R.N., Skovsen, S, Gislum, R. (2017). Designing and Testing a UAV Mapping System for Agricultural Field Surveying. *Sensors* 17(12):2703. <https://doi.org/10.3390/s17122703>
- Dusek, B. (2021). Automatic mapping of traffic signs (Közlekedési táblák automatikus térképezése). Thesis manuscript, ELTE Eötvös Loránd University, Inst. of Cartography and Geoinformatics, Budapest, Hungary.
- European Commission (2019)<sup>1</sup>. 2019/947 decree: <https://eur-lex.europa.eu/legal-content/HU/TXT/HTML/?uri=CELEX:32019R0947&from=HU#d1e2148-45-1> (last access: 2021.06.03).
- European Commission (2019)<sup>2</sup>. 2019/945 decree: <https://eur-lex.europa.eu/legal-content/HU/TXT/HTML/?uri=CELEX:32019R0945&from=HU> (last access: 2021.06.03).
- Gede, M., Árvai, V., Vassányi, G., Supka, Zs., Szabó, E., Bordács, A., Varga, Cs.G., Irás, K. (2020). Automatic vectorisation of old maps using QGIS – tools, possibilities and challenges. AUTOMATIC VECTORISATION OF HISTORICAL MAPS - International workshop organized by the ICA Commission on Cartographic Heritage into the Digital, Budapest, Hungary.
- Gede, M., Vassányi, V., Árvai, V. (2021). Experiences of MRCNN in automatized map vectorization. 15th ICA Conference Digital Approaches to Cartographic Heritage, Conference Proceedings, ISSN 2459-3893, Thessaloniki, Greece.
- Kovács, B., Pál, M., Vörös, F. (2021). Object-based image segmentation in photogrammetry for cartographic use, EGU General Assembly 2021, EGU21-11843, <https://doi.org/10.5194/egusphere-egu21-11843>.
- Kratochvílová, D., Cajthaml, J. (2020). Using the automatic vectorization method in generating the vector altimetry of the historical Vltava River Valley. *Acta Polytechnica, Czech Technical University in Prague*, 60(4):303-312.
- Lausch, A., Borg, E., Bumberger, J., Dietrich, P., Heurich, M., Huth, A., Jung, A., Klenke, R., Knapp, S., Mollenhauer, H., Paasche, H., Paulheim, H., Pause, M., Schweitzer, C., Schmulius, C., Settele, J., Skidmore, A.K., Wegmann, M., Zacharias, S., Kirsten, T., Schaepman, M.E. (2018). Understanding Forest Health with Remote Sensing, Part III: Requirements for a Scalable Multi-Source Forest Health Monitoring Network Based on Data Science Approaches. *Remote Sens.*, 10, 1120. <https://doi.org/10.3390/rs10071120>.
- Major, K., Kozma-Bognár, V., Enyedi, A., Váradi Á., Berke, J. (2016). The application of visual data of remotely controlled UAVs in education (Távirányítású drónok kutatási célú vizuális adatainak alkalmazása az oktatásban). XXII. "Multimedia in Education" international conference, Keszthely, Hungary.
- le Riche, M. (2020). Identifying Building Footprints in Historic Map Data using OpenCV and PostGIS. AUTOMATIC VECTORISATION OF HISTORICAL MAPS - International workshop organized by the ICA Commission on Cartographic Heritage into the Digital, Budapest, Hungary.
- Naaouf, N., Elek, I. (2020). Change detection analysis of land cover before and within the Syrian crises using remote sensing and geospatial techniques – a case study of Al-Salamiyah district, Syria. *Proceedings Vol. 1, 8th International Conference on Cartography and GIS, 2020, Nessebar, Bulgaria.*
- Restás, Á. (2017). Possible UAV applications in public services (A drónok közszolgálati alkalmazásának lehetőségei). *Új magyar közigazgatás. KÖZSZÖV, Gödöllő*, 10:3, pp. 49-63.
- Yang, B., Hawthorne, T.L. (2021). High-resolution UAV mapping for seagrass along the west coast of North America. *Geological Society of America Abstracts with Programs. Vol. 53, No. 2. DOI: 10.1130/abs/2021SE-362018.*