

# Lever arm calculating algorithm for efficient and precise vector determinations

**Ing. Gergely László<sup>1\*</sup>; Doc. Ing. Gábor Péter Molnár, PhD.<sup>1</sup>; Ing. Gabriella Bor<sup>2</sup>; Ing. Zoltán Tóth, PhD.<sup>1</sup>**

<sup>1</sup> Óbuda University, Institute of Geoinformatics, Alba Regia Technical Faculty, Pirosalma u. 1-3, Székesfehérvár, 8000, Hungary

<sup>2</sup> Envirosense Hungary Ltd., Department of Orthophotogrammetry, Péchy u. 46, Debrecen, 4032, Hungary

\* corresponding author: laszlo.gergely@amk.uni-obuda.hu

## ORIGINAL SCIENTIFIC ARTICLE

### ABSTRACT

In the orthophoto processing one of the key steps is the aerial triangulation during which the exterior orientation coefficients of the photographs are determined by adjustment. In order to be able to do this with sufficient accuracy, in addition to knowing the internal orientation elements of the cameras, we need the eccentricity vectors of the GNSS antennas, IMU and sensors determined in relation to each other, i.e. the lever arms. Characteristic points of the airplane and the installed instruments are measured using 3D polar method, while vectors are traditionally determined using a geometric technique. In this paper we will present an algorithm which uses an analytical geometric approach for much quicker and more precise results.

**Key words:** *Lever arm; programming; analytical geometry; photogrammetry*

## 1 INTRODUCTION

Remote sensing is used for data collection when we cannot or do not want to work directly in the area to be surveyed and also when mapping large areas. In order for a large part of the field measurements to be substituted, it is important that the accuracy of the remotely sensed data does not fall short of that available with field data collection procedures. To fulfill this aspect, our remotely sensed data must meet several criteria. For example, they are suitable if the followings are known:

- ground sample distance (GSD)
- depending on the final product, image overlap relative to each other
- sensor parameters known with sufficient accuracy
- coordinates of data collection locations determined with high precision and the spatial angle of the collected data in some known projection system, and designated direction
- and, of course, the appropriate data collecting conditions

If we are talking about photo-based aerial remote sensing, then we can ensure the desired field resolution by knowing the parameters of the measuring camera (internal orientation elements: pixel size, sensor size, focal length, principal point coordinates, distortions parameters), based on a topography model, and calculating the appropriate flight height.

Some of the required camera parameters (pixel size in millimeters, sensor size, nominal focal length) are determined by the camera and lens type, while the rest (principal point coordinates, actual focal length, distortions parameters) require camera calibration.

To plan data collection with the required overlap, in addition to the method of use of the final product (e.g.: orthophoto, true ortho), which determines the amount of overlap to be used, we also need the calibrated parameters of the camera sensor and the previously calculated flight height.

Adequate data acquisition conditions can be achieved if a sensor suitable for the purpose is used under ideal environmental conditions (weather, time of day).

Knowing the data collection locations is important because in photogrammetry - or in any remote sensing procedure - the basic task is to be able to produce field coordinates from the image coordinates of recordings, thus making it possible to carry out measurements on them. To do this, we need to determine exactly where the projection center of the camera was located in space (interpreted in a given projection) at the time of recording. If we also know the orientation of the image plane, then we know all the external orientation elements  $(x,y,z;\omega,\varphi,\kappa)$  in order to perform field coordinate calculations from pixel coordinates. [3] [4]

The problem with the above point is that the external orientation elements can only be determined indirectly, with separate sensors (GPS, IMU), not directly in the projection center or in the image plane, so the measured values are always burdened by eccentricity. For smaller, more compact systems, such as UAV survey systems, these eccentricity vectors are defined and published by the manufacturer, but for large-scale aerial photography, especially if the sensor configuration changes frequently, they need to be determined and checked regularly. [1] [2]

In this study, the authors present the traditional, geometric way of determining the eccentricity vectors, as well as the developed algorithm, which automatically enables the calculation of these vectors using an analytical method with 3d Helmert transformation.

## 2 LEVER ARM MEASUREMENT

We can perform the measurement when the sensors are installed in the aircraft. The measurement is carried out using a 3D polar method, typically with a total station.



Figure 1. 3D polar method

In order for the eccentricity vectors and the rotation angles to be properly interpreted, in addition to the sensors, a reference plane and a reference direction must also be measured. In our case, the reference plane was the base plate holding the sensors, while the direction was the longitudinal axis of the fuselage. Overall, the following points are measured:

- base plate flat, with three points (bore holes A, B, C)
- plane longitudinal axis, with two points (t1, t2)

<http://doi.org/10.51704/cjce.2023.vol9.iss1.pp35-41>

ISSN (online) 2336-7148

[www.cjce.cz](http://www.cjce.cz)

- inertial measurement unit, depending on the type, with one or two points (IMU)
- camera (fixing frame), with two points (cph1, cph2)
- hyperspectral camera, with two points (hs1, hs2)
- two GPS antennas, with one point each (ant1, ant2)

During the measurement, we stand next to the plane with the instrument in such a way that both the outer and inner points are visible at the same time, in five rounds it takes place in one face, because the errors of the axes were determined priorly, using a propped-up miniprism.

### 3 PROCESSING WITH TRADITIONAL, GEOMETRIC METHOD

We start the processing by averaging the five-round measurement. Thus, if at any point we experience an error in the measurement results, we have the option to remove such values.

The coordinate list obtained in this way is imported point by point into AutoCAD, then for easier work we also place labels next to the points with point numbers and then group them by using layer allocation.

By connecting the points of the plane, we create a plate, then we also connect the axis points so that we can use them as a reference line.

We create a user coordinate system by shifting so that the XY plane contains the plane of the base plate, and the origin coincides with one of the measured bore holes (e.g., point A).

After that, the axis-line is moved with its midpoint to the origin, then its start and end points are rotated to the XY plane by zeroing the Z coordinates.

In the next step, we rotate the +X axis in the direction of the aircraft axis so that it faces the propeller.

Using the measured IMU points and the available technical drawings, we draw the reference point of the IMU, to which we move the origin using a repeated coordinate system shift.

In this final, directional reference system, based on the sensor points, using the dimensions of blueprints and 3D models, we determine the position of the camera focal points and the center of mass of the hyperspectral camera with various editing steps.

As a final step, the X, Y, Z coordinates of the transformed points are manually read point by point and recorded in a coordinate list, which can be used to perform the trajectory processing.

### 4 PROCESSING USING ANALYTICAL METHOD

It can be seen from the previous chapter that the geometric method can be quite time-consuming and, because of its nature, is not necessarily the most accurate due to rounding and certain simplifications (e.g. we assume that the axes of the cameras are parallel to the axes of the transformed coordinate system). [5] [6]

It can also be observed that the whole process consists of displacements and rotations that can be combined into a 3D Helmert transformation (without scaling), and some edits that can be converted into polar point calculations and rotations.

Taking these into account, the algorithm is built from the following analytical geometric calculations:

The plane normal of the base plate are calculated from the coordinates of point A, B and C, (situated on the base plate) using the difference vectors of the position vectors of these points.

$$\mathbf{n} = \mathbf{v}_{AB} \times \mathbf{v}_{AC}$$

<http://doi.org/10.51704/cjce.2023.vol9.iss1.pp35-41>

ISSN (online) 2336-7148

[www.cjce.cz](http://www.cjce.cz)

This  $\mathbf{n}$  plane normal should be normalized, simply dividing by its length:

$$\hat{\mathbf{n}} = \mathbf{n}/|\mathbf{n}|$$

Using the  $t_1$  and  $t_2$  axis endpoints of the airplane, the axis vector can be expressed:

$$\mathbf{t} = \mathbf{t}_2 - \mathbf{t}_1$$

and normalizing it, we get the normalized airplane axis vector:

$$\hat{\mathbf{t}} = \mathbf{t}/|\mathbf{t}|$$

We could project this normalized airplane axis vector to the plane defined by the base plate, and we get the  $\mathbf{f}$  vector, what is on the plane and defines the forward direction.

$$\mathbf{f} = \hat{\mathbf{t}} - \hat{\mathbf{n}} \cdot (\hat{\mathbf{n}} \cdot \hat{\mathbf{t}})$$

where we use the dot product of vectors.

The cross product of the normalized  $\hat{\mathbf{f}}$  vector and the  $\hat{\mathbf{n}}$  plane normal vector is perpendicular to both of them:

$$\hat{\mathbf{p}} = \hat{\mathbf{f}} \times \hat{\mathbf{n}}$$

We need a rotational matrix, that transforms any vector to the newly defined coordinate system. This matrix would transform the  $\hat{\mathbf{p}}, \hat{\mathbf{f}}$  and  $\hat{\mathbf{n}}$  vectors to the base vectors of the newly defined coordinate system:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{R} \cdot \begin{bmatrix} \hat{\mathbf{p}}_x & \hat{\mathbf{f}}_x & \hat{\mathbf{n}}_x \\ \hat{\mathbf{p}}_y & \hat{\mathbf{f}}_y & \hat{\mathbf{n}}_y \\ \hat{\mathbf{p}}_z & \hat{\mathbf{f}}_z & \hat{\mathbf{n}}_z \end{bmatrix}$$

where  $\mathbf{R}$  matrix is simply the inverse of the right hand side matrix composed of the  $\hat{\mathbf{p}}, \hat{\mathbf{f}}$  and  $\hat{\mathbf{n}}$  vectors.

The program itself was created using Python language, as it is easy to program, extremely flexible and widespread nowadays.

Among the optional function libraries, we used numpy module for mathematical and linear algebra functions, while matplotlib module was used to plot the result.

The first step after calling the modules is to import the properly prepared coordinate file. Data lines should be Tab-separated with *ID, description, X, Y, Z* order, where the description corresponds to the previously described abbreviations. The program automatically recognizes and handles the case when the IMU was measured with two points, and then fills a list with the data, which will be used in the coming steps. [7]

In the next step, based on vectors *AB* and *AC*, the plane *ABC* is calculated, its normal vector, length and, based on these, its normal unit vector and the distance of the calculated plane from the origin.

This is followed by the calculation of a direction vector based on the axis points  $t_1, t_2$ , and then the determination of a vector in a plane, perpendicular to the axis.

After determining the length of the normal vector, we form a normal unit vector from it.

Based on the normal unit vector of the plane and the axis, we get the projection direction vector of the axis, from which, as a check, if we calculate its length, we should get units.

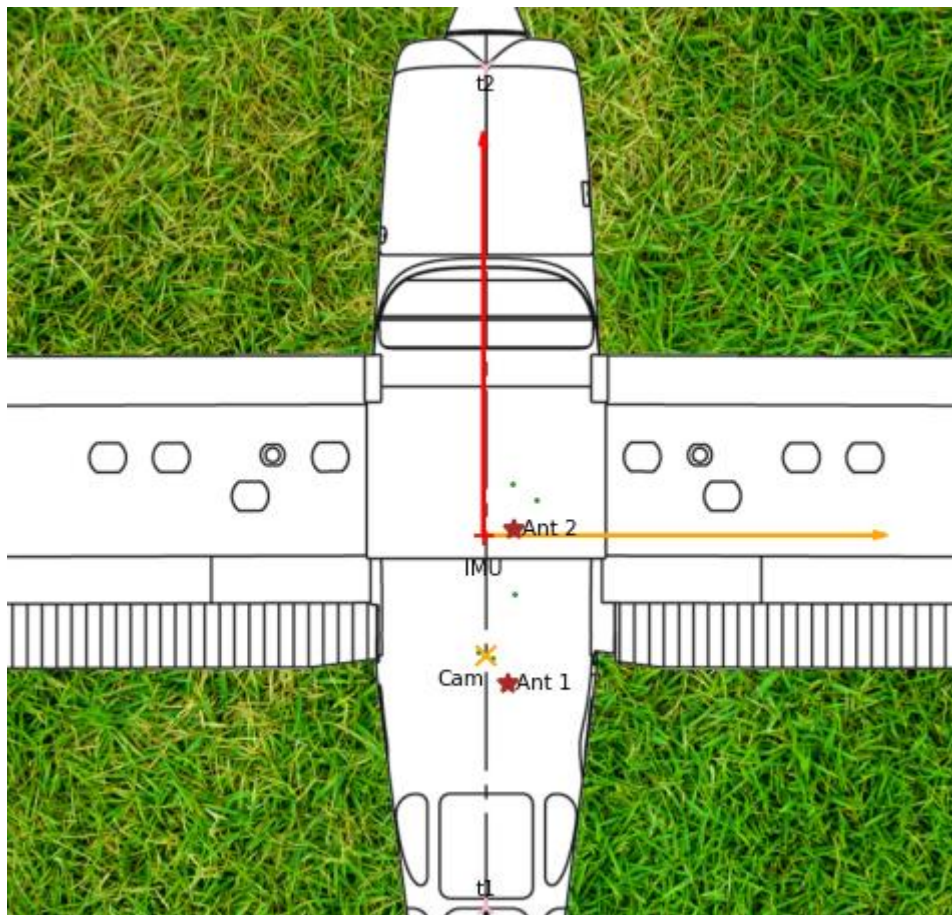
<http://doi.org/10.51704/cjce.2023.vol9.iss1.pp35-41>

Using these three unit vectors, we create the rotation matrix.

If we measured with a two-point IMU determination, the program calculates from two points, if we measured the type where only one is enough, we determine the IMU reference with a Z offset, and then perform the transformation to this point, to all measured elements, with the rotation matrix.

In the transformed system, we determine the camera focal points and the location of the center of mass of the hyperspectral sensor.

In the last step, using the top view diagram of the given airplane type, we draw the locations of the points proportionally, label them with a symbol key.



**Figure 2 Graphical results.**

In parallel with this print out the transformed coordinates of the points as a formatted character string on the screen.

```

[-3.277 -2.644 -0.825]
A [ 0.42038378  0.24041505 -0.03908122]
B [ 0.4215701  0.7561452 -0.03908122]
C [-0.09698058  0.76687788 -0.03908122]
t1 [ 0.14326238 -1.55210035  0.73907638]
t2 [0.14326238  3.29543082  0.71210277]
IMU [0.  0.  0.023]
cph1 [ 0.26852985  0.43664198 -0.0303994 ]
cph2 [ 0.05231898  0.55699464 -0.02935162]
hs1 [0.0784955  0.65341512  0.05481418]
hs2 [0.14932891  0.69373265  0.0544624 ]
ant1 [ 0.27975696 -0.25416175  0.99509714]
ant2 [0.31202478  0.63951048  1.13808918]
-222.05255335325126
Cam_b [ 0.10638737  0.49705626 -0.02935162]
Cam_j [ 0.21446146  0.49658036 -0.02935162]
Hyp COG [0.11260394  0.69651104  0.00363829]
Pont   X       Y       Z
IMU    0.0     0.0     0.0
Hyper COG  0.113  0.697  0.004
Cam b   0.106  0.497 -0.029
Cam j   0.214  0.497 -0.029
Ant e   0.312  0.64  1.138
Ant h   0.28  -0.254 0.995

```

Figure 3 Results in Python.

## 5 SUMMARY AND POSSIBLE DEVELOPMENTS

It can be said about the completed program that, since it performs the previously manual, geometrically solved task with automatic calculation, it gives results much faster, provides an immediate visual control option, and can also consider aspects in the calculation that were not possible until now due to simplifications, therefore more accurate results are obtained.

In the future, we plan for the program to be able to handle, preferably automatically recognize, the different configurations in the two planes used by the company, and to be able to compare, based on previous measurements, which points' positions have changed to an extent that exceeds the expected measurement accuracy, thus helping the possible error detection.

### References

- [1] BALÁZSIK V., TÓTH Z., ABDURAHMANOV I., 2021, Analysis of Data Acquisition Accuracy with UAV, International Journal of Geoinformatics (1686-6576 ): vol. 17 no. 1 pp.1-10.
- [2] KUGLER, Z., TÓTH, Z., SZALAY, Z., SZAGRI, D., BARSÍ, Á., 2019, Supporting microclimate modelling with 3D UAS data acquisition, Időjárás / Quarterly Journal of the Hungarian Meteorological Service, 123: 3 pp. 279-294, 16 p. <https://doi.org/10.28974/idojaras.2019.3.2>
- [3] LÁDAI, A., TOTH, C., TÓTH, Z., 2022, Indoor Mapping With An Omnidirectional Camera System: Performance Analysis, International Archives Of Photogrammetry And Remote Sensing (2002-) Xliii-b1-2022 Pp. 347-352. , 6 p. <https://doi.org/10.5194/isprs-archives-XLIII-B1-2022-347-2022>
- [4] KATONA, J., GULYAS, H. M., 2018, Determination of factors modifying land value based on spatial data In: Drótos, Dániel; Vásárhelyi, József; Czap, László; Ivo, Petráš (szerk.) Proceedings of the 19th International Carpathian Control Conference (ICCC 2018) Piscataway (NJ), Amerikai Egyesült Államok: IEEE, pp. 625-628. <https://doi.org/10.1109/CarpathianCC.2018.8399705>
- [5] LUCAS, G., HALÁSZ, L., SOLYMOSI, J., 2013, Exploring the capacities of airborne technology for the disaster assessment, HADMÉRNÖK 8 : 3 pp. 74-91. , 18 p.

- [6] LUCAS G., 2015, Considering time in orthophotography production: from a general workflow to a shortened workflow for a faster disaster response, International Archives of Photogrammetry and Remote Sensing (2002-) XL3 : W3 pp. 249-255. , 7 p.
- [7] SCHICKLER, W., THORPE, A., 1998, Operational procedure for automatic true orthophoto generation, International Archives of Photogrammetry and Remote Sensing, 32 (Part 4):527-532.