

Attitude Determination of Unmanned Aerial Vehicle using Single Camera Vector Observations

Angel Vladimirov

Faculty of computer science, University of Goce
Delchev – Stip
blvd. Krste Misirkov 10/A, 2000 Stip, Republic North
Macedonia

Saso Koceski

Faculty of computer science, University of Goce
Delchev – Stip
blvd. Krste Misirkov 10/A, 2000 Stip, Republic North
Macedonia

ABSTRACT

Technological development in the fields of electrical and mechanical engineering as well as computer and communication sciences in the last decade, have dramatically increased the popularity and fields of application of Unmanned Aerial Vehicles (UAVs). Despite the technological advancements, there are still very important challenges related to the operation of UAVs. One of the main challenging task for UAVs is to accurately determine their attitude during the flight, using the onboard sensors. This paper presents a framework for attitude determination of an UAV from single camera vector observations in a known environment. The framework has been experimentally evaluated. The results from the conducted evaluation suggest that the proposed method is appropriate and that it can be used in the control process.

Keywords

Attitude estimation, Unmanned Aerial Vehicle, Camera vector, Gauss-Newton, Levenberg-Markart.

1. INTRODUCTION

Nowadays the Unmanned Aerial Vehicles (UAV's) find a wide spectra of use in our society. UAV has initially been designed as entertainment toys but, today they are used as sophisticated information gathering tools in dangerous environments [1]. The technological advancements made the UAV's available in many different sectors. This is a result not only on the advancements in material design and their low prices but, also due to the development of novel control paradigms as well as development of Artificial Intelligence (AI). The growing popularity of the Artificial Intelligence (AI) and its application in various fields [2], starting from tourism [3], through medicine [4-6], biology [7], education [8], robotics [8-12], and also in economy [13], is mainly due to the apparatus i.e. the models and techniques used to mimic the human reasoning, learn and improve during time.

The common problem of the UAVs, even today, remains the accurate attitude determination. It is a high point of interest to know the attitude of the UAV, because of the safety of the vehicle itself and for the entire mission. For better description of this approach for solution of the attitude estimation problem, it can be assumed that every mission starts from a point in the environment that is known. This approach is based on ubiquitous visual sensor i.e. a video camera. A single camera can provide a wide source of information about the environment, and also it is a cost efficient, small and compact to attach to an UAV.

In the proposed approach the information obtained from the camera will be used as relative measurement between the UAV navigation parameters (attitude, position and velocity)

and the environment. Assuming to have a known environment, using the relative measurements, the UAV navigation parameters can be determined. One drawback of the single camera approach may be the lack of depth information.

The navigation environment could be represented in a form of a map, with defined map points. Considering the known position of the UAV obtaining the single camera position vector can be performed. Having this in mind, one may represent the navigation frame for attitude determination as a system composed of map points (environment) and the position of the UAV in the environment. In this paper two different iterative numerical methods for calculating the position vector will be investigated and compared. The first one is an iterative numerical solution based on Gauss-Newton's method and the second one is Levenberg-Marquardt method.

2. RELATED WORK

Various approaches for UAV's attitude estimation are reported in the literature. UAVs nowadays are equipped with Inertial Navigation Systems (INS). These systems usually employ Micro-Electro-Mechanical Systems (MEMS) inertial sensing technologies, which are low-cost and low-power consumption devices. The use of MEMS has become crucial part for position and movements of a UAV estimation [14]. Despite the advantages of the inertial systems based on MEMS technology they often have low precision, poor accuracy and degraded performance. Usual way for increasing their precision is to apply additional filtering and data fusion algorithms. One approach is presented in [15] where the authors are presenting fusion of data gathered from integrated gyroscopes and three-axis accelerometers.

Very interesting approach for attitude determination using the UKF and the TRIAD algorithm which include accelerometers, gyroscopes and magnetometers has been proposed in [16]. But, however the easiest and cheapest way to get a lot of information about the position and attitude of a UAV is through the visual sensor. Specific approach with single camera observations are described in [17], [18]. They are using monocular camera for gathering information and also combining the results with inertial sensors. This was actually the main point why we proceeded in this approach, asking the question what if the UAV can use only the visual sensor. The real time single camera SLAM [19], [20] method is the great step forward into detecting the position of a UAV.

3. PROPOSED METHOD

3.1 Frame setup

The orientation of a UAV in space is represented as orientation of one coordinate system (camera frame attached to the UAV) in reference to another coordinate system (a fixed coordinate system) for example World coordinate system or in this case navigation frame. The axes of these two system are connected through a linear transformation, usually it is done with transformation matrix. This matrix can be presented as a group of coordinates, like: Euler angles, quaternions, direction cosines, etc. The use of quaternions is very common in these cases, because of the linearity of the quaternions, the avoidances of usage of trigonometric functions and the small number of parameters for successful implementation. The only disadvantage of the quaternions is that they don't have a simple geometric representation, so because of that they can't be measured directly [21]. While the problem with the measurement from the single camera is the lack of precise depth data, so it may be assumed that this distance is known and constant.

For the purpose of the experiment it is assumed that the camera is mounted on the UAV on such a way that the **camera frame** is at the center of gravity of the UAV and is aligned with the **body frame** of the UAV. The camera frame is represented with $(P; x_c, y_c, z_c)$, where P stands for the center of perspective for observations, and also the center of gravity of the UAV.

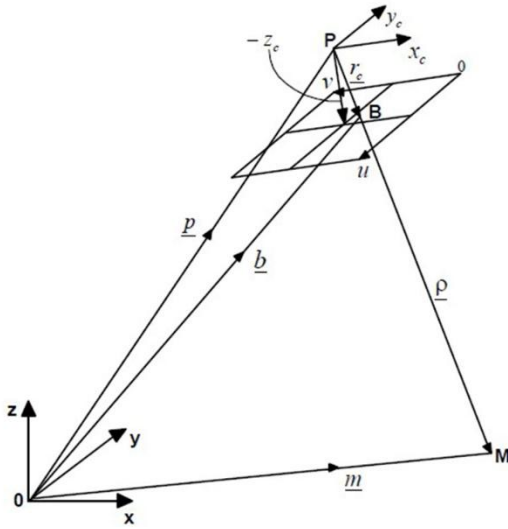


Fig 1. Camera setup vector measurements

On figure 1, two related frame systems are shown. One with center P, the center of the camera frame, and the other with center O, the center of the navigation system of the environment (coordinate system of the World). Starting from P, the axis x_c can be seen, defined in forward direction, which is perpendicular to the horizontal component u of the camera image. The axis y_c , is perpendicular with the vertical component v of the camera image. And the last axis z_c , which completes the right handed orthogonal coordinate system is perpendicular to both axis x_c and y_c . On our figure z_c has a negative sign because it is pointing to the center of the camera image and is equal to the focal length of the camera. The camera image is shown as a parallelogram with sides u and v and origin in 0.

On figure 1 in the camera frame the vector \underline{r}_c is measured. The

unit vector in the direction \underline{r}_c noted as $\underline{\hat{r}}_c$ may be calculated as

$$\underline{\hat{r}}_c = \frac{\underline{r}_c}{|\underline{r}_c|} \quad (1)$$

Vector $\underline{\hat{r}}_c$ can be transformed in the navigation frame with the transformation matrix, C_{b2n} , that transforms body frame coordinates into navigation frame coordinates.

The transformation process is explained in the following:

$$\underline{r} = C_{b2n}\underline{r}_c \quad (2)$$

In the last equation, the transformation matrix C_{b2n} is given in quaternion form:

$$C_{b2n} = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} \quad (3)$$

After the transformation for vector \underline{r} , the following applies:

- $\underline{r} = \underline{b} - \underline{p}$
- $\underline{\hat{r}} = \frac{\underline{b} - \underline{p}}{|\underline{b} - \underline{p}|}$
- $\underline{\rho} = \mu \underline{\hat{r}}$, where μ is an unknown parameter.

The $\underline{\rho}$ is the difference between the map point vector \underline{m} and the position vector \underline{p} and may be written as

$$\underline{\rho} = \underline{m} - \underline{p} \quad (4)$$

The unit vector in the direction $\underline{\rho}$ noted as $\underline{\hat{r}}$ in the navigation frame also may be evaluated as

$$\underline{\hat{r}} = \frac{\underline{\rho}}{|\underline{\rho}|} \quad (5)$$

The quaternion vector $\underline{q} = [q_0 \ q_1 \ q_2 \ q_3]^T$ used in the transformation matrix is with unit normalization

$$|\underline{q}| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \quad (6)$$

The advantage of the quaternions is that they could be used for a transformation of the unit vector $\underline{\hat{r}}_c$ in the camera frame system, into a unit vector $\underline{\hat{r}}$ in the navigation frame:

$$\underline{\hat{r}} = C_{b2n}\underline{\hat{r}}_c \quad (7)$$

It is assumed that the data about the vector \underline{m} , vector \underline{p} from where the measurement is taken and the position where the single camera is mounted, are known. Measurements and observations are conducted with the camera to the map points, in order to find out the attitude of the UAV (camera) in accordance with the environment, like in figure 1.

Actually the quaternion vector $\underline{q} = [q_0 \ q_1 \ q_2 \ q_3]^T$ is going to be computed. With the measurements and observations from the camera, the unit vector $\underline{\hat{r}}_c$ is computed in the camera frame system, equation (1). Using the vectors \underline{m} and \underline{p} the depth $\underline{\rho}$ can be determined with the equation (4). Next the depth $\underline{\rho}$ it is used in the equation (5) in order to compute the unit vector $\underline{\hat{r}}$ in the navigation frame system. Now that the two vectors are known, one in the camera frame, and the other in the navigation frame, the quaternion vector \underline{q} can be determined. The equation (7) in matrix form can be written like this:

$$\begin{bmatrix} \hat{L}_x \\ \hat{L}_y \\ \hat{L}_z \end{bmatrix} = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} \begin{bmatrix} \hat{L}_{xx} \\ \hat{L}_{yy} \\ \hat{L}_{zz} \end{bmatrix} \quad (8)$$

Since the quaternion vector \underline{q} is a four element vector, additional vector should be added to equation (8) in order to calculate the four unknowns in \underline{q} . The property of the quaternion vector to be unit normalized can be added to the equation, i.e. equation (6) can be added to equation (8), and system of nonlinear equations is formed:

$$\begin{cases} \hat{L}_x = (q_0^2 + q_1^2 - q_2^2 - q_3^2)\hat{L}_{xx} + 2(q_1q_2 - q_0q_3)\hat{L}_{xy} + 2(q_1q_3 + q_0q_2)\hat{L}_{xz} \\ \hat{L}_y = 2(q_1q_2 + q_0q_3)\hat{L}_{xx} + (q_0^2 - q_1^2 + q_2^2 - q_3^2)\hat{L}_{yy} + 2(q_2q_3 - q_0q_1)\hat{L}_{zz} \\ \hat{L}_z = 2(q_1q_3 - q_0q_2)\hat{L}_{xx} + 2(q_2q_3 + q_0q_1)\hat{L}_{yy} + (q_0^2 - q_1^2 - q_2^2 + q_3^2)\hat{L}_{zz} \\ 1 = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \end{cases} \quad (9)$$

In order to solve this system of four nonlinear equations, two approaches are used, Gauss-Newton and Levenberg-Markart method.

The nonlinear system of equations (9) can be written as:

$$Y_n = f(x_n, \theta) + Z_n \quad (10)$$

The approach that Gauss suggest is to use a linear approximation to the nonlinear function to iteratively improve an initial guess θ^0 for θ and keep improving the estimates until there is no change [4]. Expanding the nonlinear function $f(x_n, \theta)$ in first order Taylor series about θ^0 as $f(x_n, \theta) = f(x_n, \theta^0) + v_{n1}(\theta_1 - \theta_1^0) + v_{n2}(\theta_2 - \theta_2^0) + \dots + v_{nn}(\theta_n - \theta_n^0)$ where

$$V_{nn} = \frac{\partial f(x_n, \theta)}{\partial \theta_n} |_{\theta^0} \quad n = 1, 2, \dots, N. \quad (11)$$

and if all N cases are included, it can be written as

$$\eta(\theta) \approx \eta(\theta^0) + V^0(\theta - \theta^0) \quad (12)$$

where V^0 is the N x N derivate matrix with elements $\{V_{nn}\}$. This is equivalent to approximating the residuals, $z(\theta) = y - \eta(\theta)$, by

$$z(\theta) = y - [\eta(\theta^0) + V^0\delta] \quad (13)$$

where $z^0 = y - \eta(\theta^0)$ and $\delta = \theta - \theta^0$. Then the Gauss increment δ^0 can be calculated to minimize the approximate residual sum of squares $|z^0 - V^0\delta|$ using

$$V^0 = QR = Q_1R_1 \quad (14)$$

$$w_1 = Q_1^T z^0 \quad (15)$$

$$\eta^1 = Q_1 w_1 \quad (16)$$

and so

$$R_1 \delta^0 = w_1 \quad (17)$$

The point

$$\eta^1 = \eta(\theta^1) = \eta(\theta^0 + \delta^0) \quad (18)$$

now should be closer to y than $\eta(\theta^0)$, and it can be moved to this better parameter value $\theta^1 = \theta^0 + \delta^0$ and perform another iteration by calculating new residuals $z^1 = y - \eta(\theta^1)$, a new derivative matrix V^1 , and a new increment. This process is repeated until convergence is obtained, that is, until the increment is so small that there is no change in the elements of the parameter vector. Convergence implies that the best estimates of the parameters are obtained. The Gauss Newton method provides solution even if the system may not have a

zero i.e. returns a point where the residual is small. If the Jacobian of the system is singular the Gauss Newton method might converge to a point that is not a solution of the system of equations [22].

Levenberg-Markart method is the second approach that it is used in this paper. This approach represents a slight modification to the Gauss-Newton method and it has been used widely in the computer vision literature [23]. The main difference from the Gauss-Newton method is to set $\alpha^i = 1$ and use instead $D^i = (\lambda^i I + \nabla u(x^i) \nabla u(x^i)^T)^{-1} \in \mathbb{R}^{n \times n}$

where $\lambda^i > 0$ is a scalar determined by the following rules, initially it is assigned to small values, and then:

1. If the current value of λ results in a decrease in the error, then the iteration is accepted and λ is divided by 10 as the initial value for the iteration.
2. If λ results in an increase in the error, then it is multiplied by 10, and the iteration is tried again until λ is found that results in a decrease in the error.

Because of this, the Levenberg-Marquardt method still works even if the Jacobian matrix is not of full rank, which occurs often in practice. The best description is that the method tends to adapt its step size through controlling the value of λ , based on the history of values of the objective function.

3.2 Camera calibration

Geometric camera calibration can be used for estimation of the parameters of a lens and image sensor of an image or video camera. Afterwards these parameters can be used to correct the lens distortion, measure the size of an object in world units, or determine the location of the camera in the space.

Camera parameters include intrinsic, extrinsic, and distortion coefficients. To estimate the camera parameters, a 3-D world points are needed and their corresponding 2-D image points. These correspondences can be obtained by using multiple images of a calibration pattern, such as a checkerboard.

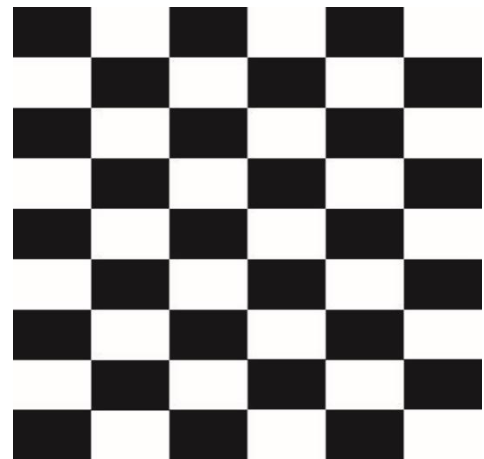


Fig 2. Checkerboard pattern for camera calibration

The calibration pattern as it can be seen in figure 2 is a rectangle, with different number of squares on a side. We have to know the dimensions of each square of the pattern and also the dimension of the whole pattern. All this information is needed in the software for calibration. The pattern has to be printed and afterwards 10-20 pictures are taken of it from different angles. Through the calibration software the parameters are obtained. The calibration algorithm calculates

the camera matrix using the extrinsic and intrinsic parameters. The extrinsic parameters represent a rigid transformation from 3-D world coordinate system to the 3-D camera's coordinate system. The intrinsic parameters represent a projective transformation from the 3-D camera's coordinates into the 2-D image coordinates.

The intrinsic parameters are the one that are needed because they include the focal length, the optical center, and the skew coefficient. The camera intrinsic matrix, K , is defined as:

$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix} \quad (19)$$

$[c_x, c_y]$ - Optical center in pixels.

(f_x, f_y) - Focal length in pixels.

$$f_x = \frac{F}{p_x} \quad (20)$$

$$f_y = \frac{F}{p_y} \quad (21)$$

F - Focal length in world units, typically expressed in millimeters.

(p_x, p_y) - Size of the pixel in world units.

$s = f_y \tan \alpha$ - Skew coefficient.

3.3 Quaternions expressed in terms of Euler angles

In order to get the better representation of the UAV's attitude, the quaternions can be expressed in terms of Euler angles, which are easier to understand at the end of the calculation of the system of nonlinear equations (9).

$$q_0 = \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \quad (22)$$

$$q_1 = \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \quad (23)$$

$$q_2 = \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \quad (24)$$

$$q_3 = \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \quad (25)$$

And this is the opposite expression of Euler angles into quaternion form:

$$\text{Roll angle: } \phi = \arctan \left[\frac{2(q_2 q_3 + q_0 q_1)}{(q_0^2 - q_1^2 - q_2^2 + q_3^2)} \right] \quad (26)$$

$$\text{Pitch angle: } \theta = \arcsin [-2(q_1 q_3 - q_0 q_2)] \quad (27)$$

$$\text{Yaw angle: } \psi = \arctan \left[\frac{2(q_1 q_2 + q_0 q_3)}{(q_0^2 + q_1^2 - q_2^2 - q_3^2)} \right] \quad (28)$$

4. EXPERIMENTAL EVALUATION AND RESULTS

An experimental environment has been set up for the practical tests with dimensions: 2 meters along x-axis, 3 meters along y-axis and 0 meters along z-axis. The experimental environment and the coordinate center of the navigation system are shown on figure 3.

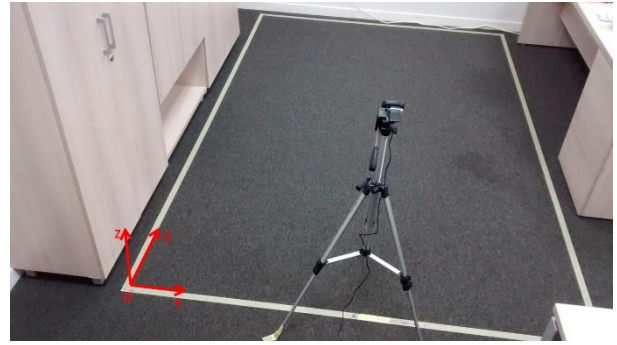


Fig 3. Experimental environment and navigational system

Camera Logitech C270 is mounted on a camera stand. The camera has focal length of 4 mm. With the process of camera calibration mentioned in 3.2, the pixel size can be calculated, which is 0.0070622 mm for each pixel. Three short videos are taken with the camera, at a frequency of 15Hz with pixel resolution 640x480 and duration of 20 seconds.

In the first video the yaw angle of the camera is changed, in the second the pitch angle of the camera and in the third video the roll angle of the camera is changed.

With precise measurements the initial vector from the camera position is computed, $\underline{p} = [1.0285m \ 0.05m \ 1m]^T$. The initial Euler angles given in degrees are precisely determined: roll angle $\phi = 0$, pitch angle $\theta = -3.5$ and yaw angle $\psi = 0$.

Three small circles with radius of 50 mm are set on the floor in the experimental environment. They are used as map (reference) points for attitude determination of the camera. The position of each circle is known and they are presented on figure 4 as: $\underline{m}_1 = [0.5m \ 1.5m \ 0m]^T$, $\underline{m}_2 = [1m \ 2.25m \ 0m]^T$, $\underline{m}_3 = [1.5m \ 1.5m \ 0m]^T$.

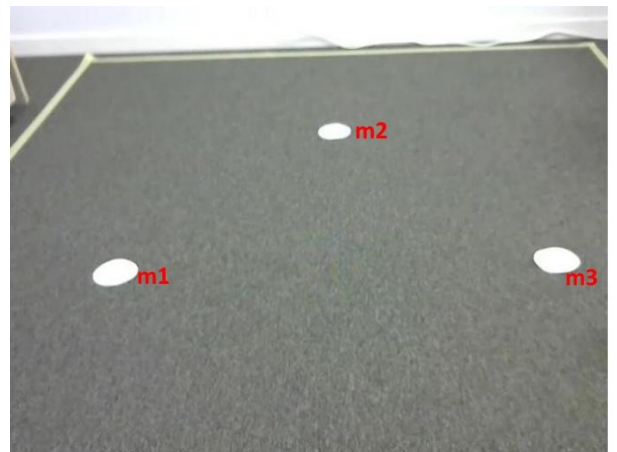


Fig 4. Three circles in the work environment

The software that is being used is Matlab. In Matlab there is a package called Optimization toolbox, which includes ready to use optimization functions. In this case the function lsqnonlin is used. This function enables solving problems such as nonlinear least-squares problems and nonlinear data-fitting problems. The function lsqnonlin needs defining of a user functions for determining parameters in vector form. This is the basic form for optimization problem that can be solved with the use of lsqnonlin:

$$F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \end{bmatrix} \quad (29)$$

In the last formula, x is vector or matrix and $f(x)$ is a function/s which returns vector or matrix values. For example written in Matlab

$$x = lsqnonlin(fun, x0) \quad (30)$$

`lsqnonlin` begins from point x_0 and finds the solution described with the function fun .

Setting the option,

```
options = optimset('LevenbergMarquardt', 'on');
```

the Levenberg-Marquardt method is set up and it has initial value $\lambda = 0.01$. With additional setting in `options.LevenbergMarquardt` into 'off' and `options.LargeScale` into 'off' the Gauss Newton method is set up.

After all this is done the algorithms are executed, actually instead of processing the full videos, a few images are separated from each video and set for processing in the algorithms. This is time and cost efficient and still capable to calculate the results.



Fig 7. Change of Pitch angle, before image processing.

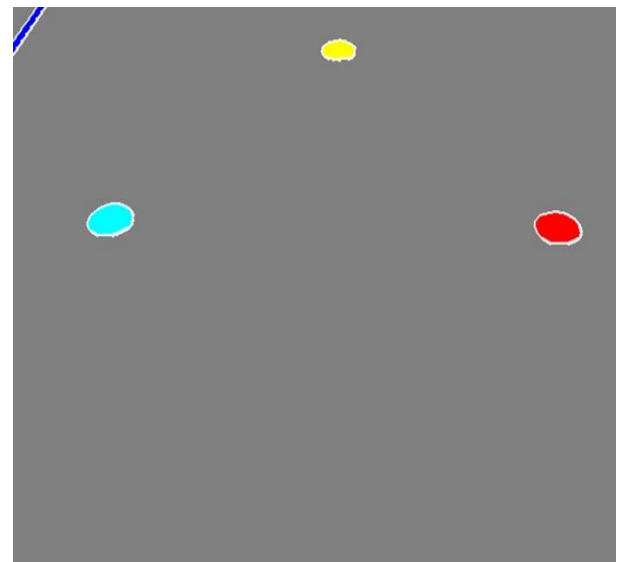


Fig 8. Change of Pitch angle, after image processing.



Fig 5. Change of Yaw angle, before image processing.

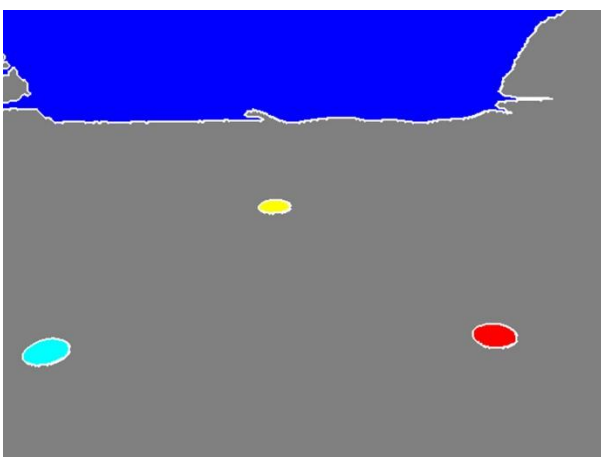


Fig 6. Change of Yaw angle, after image processing.



Fig 9. Change of Roll angle, before image processing.

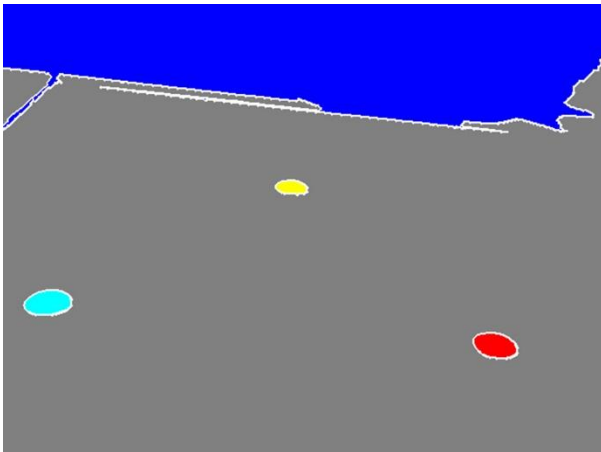


Fig 10. Change of Roll angle, after image processing.

With both of the methods the new Euler angles of the UAV have been successfully calculated, actually the new attitude of the UAV has been determined after the simulated movement.

Table 1. Comparison of the initial values and the new calculated values of the quaternions

	q0	q1	q2	q3
Initial (Fig 4)	0.9995 335912 00424	0	- 0.03053 8501305 477	0
Fig 6	0.9108	0.3526	-0.0063	-0.0664

One example is shown in detailed values before and after simulated movement. Initial and new calculated values for quaternions are shown in Table 1, but as mentioned earlier the representation of the quaternions itself doesn't give much explanation of the attitude of the UAV or of the changes made on it. In order to see the difference in both situations, quaternions are expressed in terms of Euler angles, which are easier to understand and visualize, Table 2.

Table 2. Comparison of the initial values and the new calculated values of the Euler angles

	Yaw	Pitch	Roll
Initial (Fig 4)	0	-3.5	0
Fig 6	-0.1313	0.0370	0.7362

The idea of the movement in Figure 6 was to make simulation of attitude change of the UAV especially regarding the Yaw angle. The results that came after executing both of the algorithms, Gauss-Newton and Levenberg-Marquardt, for Figure 6 compared to the initial attitude of the UAV, Figure 4 were not so simple. It can be seen that by moving the UAV only in direction relevant around the vertical axis with purpose to change the Yaw angle, changes are made in all three Euler angles. With this experiment it is emphasized that all Euler angles must be taken in consideration for attitude determination of the UAV.

5. CONCLUSION

Attitude determination of UAV in space is a difficult issue for solving. In this paper it was presented an approach for attitude determination of a UAV with the use of vector observations from a single camera into a known environment. Firstly through a function for circle detection in Matlab, the 3 circles were detected in the work environment, which location was known. So after that with the image processing, the appropriate solution was obtained for the attitude determination of the UAV. The solutions were calculated with the use of iterative numerical methods Gauss-Newton and Levenberg-Marquardt method. Quaternions were used, because of their linearity and the possibility of avoiding calculating trigonometric functions, as well as the small number of parameters needed for successful implementation and of course short time for processing. The results that came up from the 2 methods were compared and they show that the Levenberg-Marquardt method is the better solution for this problem. Actually the Levenberg-Marquardt method is an improved modification of the Gauss-Newton method and in these days finds wide specter of use in the literature of computer vision.

6. REFERENCES

- [1] Z. Mohajerani., 'Vision-based UAV pose estimation' A Thesis at Department of Electrical and Computer Engineering, Northeastern University Boston, Massachusetts 08/2008.
- [2] Loshkovska, Suzana, and Saso Koceski, eds. ICT innovations 2015: Emerging technologies for better living. Vol. 399. Springer, 2015.
- [3] Petrevska, Biljana, and Saso Koceski. "Recommending ideal holiday at national level." Journal of Applied Economics and Business 1, no. 1 (2013): 15-22.
- [4] Kulev, Igor, Elena Vlahu-Gjorgievska, Vladimir Trajkovik, and Saso Koceski. "Development of a novel recommendation algorithm for collaborative health: Care system model." Comput. Sci. Inf. Syst. 10, no. 3 (2013): 1455-1471.
- [5] Stojanov, Done, and Saso Koceski. "Topological MRI prostate segmentation method." In Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, pp. 219-225. IEEE, 2014
- [6] Koceski, Saso, and Natasa Koceska. "Evaluation of an assistive telepresence robot for elderly healthcare." Journal of medical systems 40, no. 5 (2016): 121.
- [7] Stojanov, Done, Aleksandra Mileva, and Sašo Koceski. "A new, space-efficient local pairwise alignment methodology." Advanced Studies in Biology 4, no. 2 (2012): 85-93.
- [8] Koceski, Saso, and Natasa Koceska. "Challenges of videoconferencing distance education-a student perspective." International Journal of Information, Business and Management 5, no. 2 (2013): 274.
- [9] Koceski, Saso, Natasa Koceska, and Ivica Koccev. "Design and evaluation of cell phone pointing interface for robot control." International Journal of Advanced Robotic Systems 9, no. 4 (2012): 135.
- [10] Koceski, Saso, Stojanche Panov, Natasa Koceska, Pierluigi Beomonte Zobel, and Francesco Durante. "A novel quad harmony search algorithm for grid-based path

- finding." *International Journal of Advanced Robotic Systems* 11, no. 9 (2014): 144.
- [11] Koceska, Natasa, Saso Koceski, Francesco Durante, Pierluigi Beomonte Zobel, and Terenziano Raparelli. "Control architecture of a 10 DOF lower limbs exoskeleton for gait rehabilitation." *International Journal of Advanced Robotic Systems* 10, no. 1 (2013): 68.
- [12] Serafimov, Kire, Dimitrija Angelkov, Natasa Koceska, and Saso Koceski. "Using mobile-phone accelerometer for gestural control of soccer robots." In *Embedded Computing (MECO), 2012 Mediterranean Conference on, Bar, Montenegro*, pp. 140-143. 2012.
- [13] Koceska, Natasa, and Saso Koceski. "Financial-Economic Time Series Modeling and Prediction Techniques—Review." *Journal of Applied Economics and Business* 2, no. 4 (2014): 28-33.
- [14] Kada, B., K. Munawar, M. S. Shaikh, M. A. Hussaini, and U. M. Al-Saggaf. "UAV attitude estimation using nonlinear filtering and low-cost mems sensors." *IFAC-PapersOnLine* 49, no. 21 (2016): 521-528.
- [15] Gu, Yu, Jason N. Gross, Matthew B. Rhudy, and Kyle Lassak. "A fault-tolerant multiple sensor fusion approach applied to UAV attitude estimation." *International Journal of Aerospace Engineering* 2016 (2016).
- [16] A. Rodriguez-Ramos., C. Sampedro., A. Carrio and H. Bavl 'A Monocular Pose Estimation Strategy for UAV Autonomous Navigation in GNSS-denied Environments' *Computer Vision Group, Center for Automation and Robotics (UPM-CSIC) - Project MICYT DPI2014-60139-R.*
- [17] S. P. Won, F. Golnaraghi, and W. W. Melek, "A fastening tool tracking system using an imu and a position sensor with kalman filters and a fuzzy expert system," *IEEE Transactions on Industrial Electronics* , vol. 56, no. 5, pp. 1782–1792, May 2009.
- [18] D. Du, L. Liu, and X. Du, "A low-cost attitude estimation system for uav application," in *Chinese Control and Decision Conference (CCDC)* , May 2010, pp. 4489–4492
- [19] Hector Garcia de Marina - IEEE, Fernando J. Pereda, Jose M. Giron-Sierra-IEEE, and Felipe Espinosa-IEEE, "UAV attitude estimation using Unscented Kalman Filter and TRIAD" arXiv:1609.07436v1 [cs.RO] 23 Sep 2016.
- [20] S. Ahrens, D. Levine, G. Andrews, and J. P. How, "Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference, 2009*, pp. 2643–2648.
- [21] A. D. Wu, E. N. Johnson, M. Kaess, F. Dellaert, and G. Chowdhary, "Autonomous flight in gps-denied environments using monocular vision and inertial sensors," *Journal of Aerospace Information Systems*, 2013.
- [22] D.M.Bates and D.G.Watts,'*Nonlinear Regresion Analysis and its Applications*', John Wiley and Sons Inc.,1988.
- [23] Y. Ma, S. Soatto, J. Kosecka, S. Sastry "Invitation to 3-D Vision: From images to geometric models", Springer 2010.