UDC 65.011.56:633.1(045)
DOI:10.18372/1990-5548.62.14386

[1]V. M. Sineglazov,
[2]V. S. Ischenko

ALGORITHMIC SUPPORT OF THE VISUAL NAVIGATION SYSTEM

[1,2]Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation, Electronics and Telecommunications, National Aviation University, Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua ORCID 0000-0002-3297-9060, [2]IschenkoVitaly@gmail.com

***Abstract*—*Article discusses the implementation of software architecture and algorithmic support for a visual navigation system on-board computer, which imposes restrictions on computing power in the implementation. Partitioning algorithms by logic is proposed, thus logic splitting by layers and can be processed separately on different hardwares. Also in this papper overview of obtaining gps coordinates from pixel shifting is presented. Simple runtime processes and diagnostic operational visual navigation system module is given.*

Index Terms—Visual navigation system; convolutional neural network; computer vision; local features of the image; detectors; descriptors of local features; layers of software architecture; operating system processes; satellite navigation system; inertial navigation system.

I. INTRODUCTION

On modern UAVs, the inertial navigation system (INS), in particular the strapdown inertial system (SINS), is used as the main navigation system, which provides the calculation of the coordinates of the location and orientation angles of the UAV. However, SINS has the ability to accumulate error. These INS properties put forward stringent requirements for the accuracy of primary information sensors (gyroscope, accelerometer, magnetometer), most of which cannot be technically implemented at a given time. Therefore, the INS requires correction, which is provided by the satellite navigation system (SNS).

Today, UAVs are increasingly used for aerial reconnaissance and are in the area of electronic warfare (EW), thus blocking the signals of the SNS. Full dependence of navigation on the SNS, even in flight mode on a flight in the presence of interference from electronic warfare equipment, makes it impossible to complete the flight tasks and the loss of UAVs.

To solve this problem, additional sources of navigation information of a high degree of noise immunity are needed. As such an additional source of navigation information, a visual navigation system (VNS) is proposed.

In this paper, proposed an approach to the implementation of software architecture and algorithms for the VNS module.

II. PROBLEM STATEMENT

Based on data from primary information sensors (accelerometers and gyroscopes, altimeter, magnetometer), visual information coming from the camera, the following is necessary:

– ensure the correct orientation of the camera in relation to the ground;
– provide camera control;
– estimate pixel offset between two image frames;
– evaluate the current UAV coordinates in accordance with the specified accuracy criterion.

The accuracy criterion of the SVN is calculated by the formula: $\mathcal{L} = \Omega + \alpha$, where $\mathcal{L}$ is the total accuracy criterion of the SVN module Ω is the accuracy of the GPS receiver at the take-off point of the UAV α is the maximum permissible error in estimating GPS coordinates in centimeters.

III. SUBSTANTIATION OF THE CHOICE OF SOFTWARE ARCHITECTURE AS LAYERS

Development, selection of parameters of visual navigation algorithms, accuracy assessment of several algorithms running in parallel on the on-board computer of the visual navigation module is resource-intensive, long-time and not convenient. Due to the limited power of the on-board processor, a layer-by-layer processing architecture was chosen. The first layer of the visual navigation module is designed to prepare, normalize and transfer source images (transmit data to the next layers), which are located and running on another powerful PC – laptop for processing and evaluating the results. Other layers are separate under the programs and simultaneously receive the initial data from the first layer. Next, each layer performs its processing, displays the results, graphs of accuracy comparisons of several approaches for determining UAV coordinates at once. The most accurate and stable of the algorithms based on their accuracy results are subsequently built into the on-board computer of the visual navigation module. The process of training a convolutional neural network model for the task of detecting, classifying, and segmenting is hundreds of times faster on a stationary powerful PC where this layer is running, which significantly saves the time for developing and evaluating the accuracy of visual navigation algorithms.

In accordance with the task, the structural diagram (Fig. 1) of the software has the following form (Fig. 2).

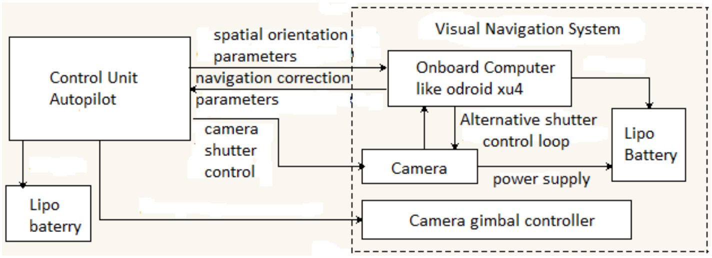

Fig. 1. Structure diagram of visual navigation system

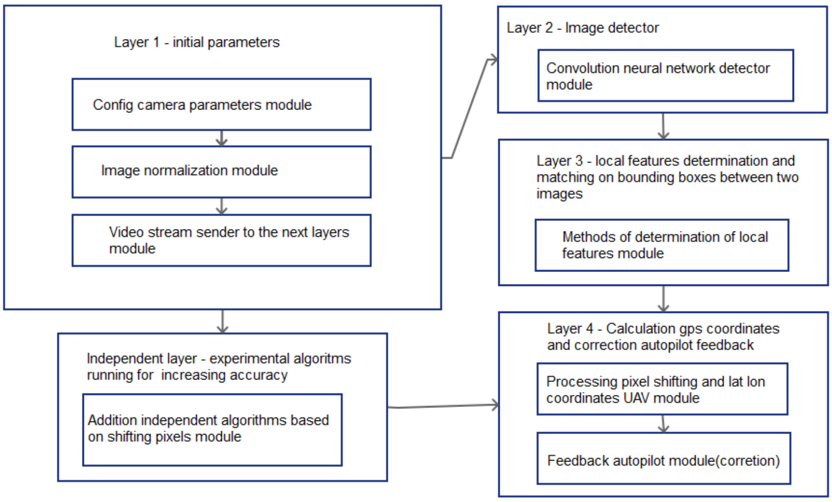

Fig. 2. Software architecture of visual navigation system

III. ALGORITHMIC SOFTWARE LAYERING (COROUTINES SOFTWARE MODULES) VNS

1) The image pre-processing layer includes the following routines:

Camera settings subroutine – sets the basic parameters for photographing for further processing and normalization.

Settable camera parameters:

– frame size, for example, 1080x720;

– the period of photographing (based on the cruise speed of the UAV);

– percentage of image overlap (50–60% recommended);

– camera matrix size and focal length;

– pixel scale factor on the ground (in centimeters);

– the height at the take-off point at which you need to fix the GPS signal and attach it to the center of the image.

Image normalization subroutine – provides invariance to changes in illumination between images: In most cases, to restore the color characteristics of an unevenly illuminated image, the following is done: linear correction of brightness and contrast, power (gamma) correction, and, logarithmic correction (retinex).

In this subroutine, the Multi-ScaleRetinex algorithm + contrast correction is used to normalize images.

Subroutine to send the initial normalized video stream to the following layers – sends normalized images received from the camera to all layers, this is necessary for parallel processing by each layer of its image section with certain algorithms to solve the navigation problem in the following layers.

Each layer represents a separate client-server application host and has the following routing scheme (Fig. 3).

Also, this software module provides framing of the video stream, its backup recording into memory and conversion to various photo / video formats (jpeg, png, jpg, bmp, mp4, avi, mkv and others).

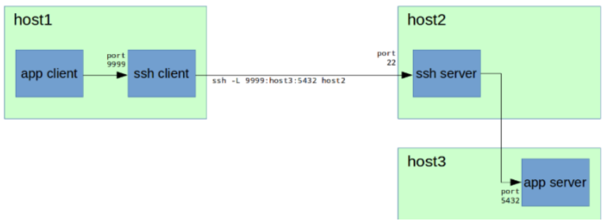

Fig. 3. Scheme of data routing between layers

2) Image processing layer for searching and mapping local features.

This layer is the most important in the visual navigation system and the accuracy of the VNS and the accuracy of determining the coordinates of the UAV as a whole depend on the accuracy of its algorithms.

Convolutional Neural Network Detector Subroutine – based on the pre trained model, it determines the most unique areas in the image or known objects, segments them and transfers the pixel coordinates (x, y) of the areas to the next layer.

The convolutional neural network in this layer is used to determine informal features in the image and their further detection.

Subprogram of methods for determining local features in the image areas received at the input of this layer.

The algorithm is implemented using the computer vision library (OpenCV) and has the following procedures.

2.1) Determination of local features in the image and the creation of descriptors for many points.

Descriptors must meet the following requirements:

– be specific (different points must have different descriptors);

– be local, as well as points (for a point, the description should not depend on the whole image, but only on a small neighborhood);

– invariant to image distortions (it is necessary to find points in the same region of the image regardless of its distortions, it would be correct if the handle of such a point does not change during its geometric transformations);

– it is desirable that the descriptors are quickly processed on the on-board processor.

2.2) Mapping points of local features in two images.

Problem statement – there are local features on two images described by descriptors, it is necessary to compare the points with each other.

Solution – candidate pairs are generated, each point (local feature) of one image is compared with all points of the features of the second image and the best pairs are selected according to the selected metric.

A simple option is exhaustive search, since exhaustive search is very slowly using various approximate metrics. For example, it is possible to build a hierarchical structure for this, all descriptors are taken and a cd-tree is built from them. Or, for example, the hash function is used.

The next step - the generated set of candidate pairs must be filtered using any heuristic features. Often a situation arises if there are a lot of similar areas on one image, then it is impossible to find the perfect pair for our point, there will be many candidates, and the goal is to use only bright features, for example, for a building (top view) in the image to use only such bright features as the roof.

Therefore, there remains the only opportunity for each feature point in the first image to find several best candidates, for example, two in the second image and compare them in quality, if the quality is about the same, then this point is not reliable and it is discarded. If the quality ratio of the first pair and the second pair is high – more than 0.8, then this point is more unique and it will be used for comparison. Such a mapping of points the metric is asymmetric, therefore it is necessary to build a symmetric metric from this, the standard symmetric metric is the MND (mutual neighbor distance) metric, which is the mutual nearest distance.

3) Layer of calculation from the pixel offset of the current GPS coordinates of the UAV.

To calculate the coordinates of significant points on the reference image at the take-off point, it is necessary to determine the pixel distance from the center of the image, the coordinates of which we know, to the points filtered by the RANSAC method.

Pixel distance (offset) d between points $A(x_1, y_1)$ and $B(x_2, y_2)$ on the plane is determined by the equation

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

The resulting distance is equal to the number of pixels, which must be multiplied by the current scale of the camera matrix in meters at a known height and focal length.

The tangent of the angle between the segment and the positive direction of the Ox axis is determined by the formula (this angle is counted from the Ox axis counterclockwise)

$$\operatorname{tg} \varphi = \frac{y_2 - y_1}{x_2 - x_1}.$$

The tangent determined by this formula is the angular coefficient of the line. In addition, it is necessary to attach to each image a compass course read from the autopilot.

Having determined the distance to the points and the angle, it is necessary to recalculate the coordinates of their latitude and longitude, considering them to be the destination points relative to the central point of the image, the distance from which we know by the following equation:

$$\varphi_2 = \arcsin\left(\sin\varphi_1 \cos\delta + \cos\varphi_1 \sin\delta \cos\theta\right),$$
$$\lambda_2 = \lambda_1 + \operatorname{arc\,tg} 2\left(\sin\theta \sin\delta \cos\varphi_1,\ \cos\delta - \sin\varphi_1 \sin\varphi_2\right),$$

where, φ_1 is the current longitude, φ_2 is the longitude of the target point, the mission loaded in the UAV autopilot memory, λ_1 is the current latitude, λ_2 is the latitude of the target point, R is the radius of the Earth (average radius is 6.371 km).

The center of the first image at the take-off point is taken as the reference point of the origin. After that, the found landmarks in the image will be associated with this point and its GPS coordinates. By extrapolating the obtained coordinates from the images, and reading from the autopilot memory the recorded coordinates of the points of the auto mission of the flight plan, we obtain the coordinates of the target point and the current one. Using the haversine equation, we consider the distance to the target point through the arc of a large circle approximating the Earth as a sphere in meters.

For some autopilots, it is also necessary to calculate the course to the target point for navigating the equation:

$$\theta = \operatorname{arc\,tg} 2\left(\sin\Delta\lambda \cos\varphi_2,\ \cos\varphi_1 \sin\varphi_2 - \sin\varphi_1 \cos\varphi_2 \cos\Delta\lambda\right),$$

where, φ_1 λ_1 is the longitude / latitude of the current point, φ_2 λ_2 is the latitude / longitude of the calculated point (destination) of the mission loaded from the UAV autopilot memory.

IV. Diagnostic of Operational VNS

Processing of navigation data by the visual navigation module is performed sequentially through the layers of the proposed software architecture. Each layer is a separate process in the Linux Debian operating system, when power is applied to the module, the main process starts at startup, which starts all other processes (layers), the main process has the identifiers of each child process and receives messages about the successful start of all processes (layers), which sends to the lcd matrix of the module for visualization.

Subsidiary processes are started in turn and the main process sequentially writes messages about the successful launch of each of the processes on the LCD screen, which is located in the housing of the SVN module. Upon successful start of all child processes, in addition to the text, the green LED lights up (Fig. 4).

Fig. 4. General view of visual navigation system

The linux debian operating system used on the on-board computer of the SVN module allows you to automatically restart the main and all child processes in the event of their abnormal termination or unsuccessful start.

There is also a set of buttons and LEDs including an LED matrix for configuring and debugging the visual navigation module, details will be given in the following articles.

V. Conclusion

There are substantiated the necessity of creating visual navigation system for UAV basis on distributed by layers software architecture, due to onboard computer processing and dimensions limits. Was shown that software architecture and description of all processing layers, main tasks and algorithms.

References

[1] V. A. Rogoshyn, V. M. Sineglazov and N. K. Filyashkin, *Flight Navigation Complexes of Aircrafts*. Kyiv, NAU, 2004, 238 p. (in Ukrainian).

[2] Flight controller Pixhawk http://pixhawk.org

[3] Paul D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Second Edition (Artech House Remote Sensing Library) 2nd Edition, 2008, 503 p.

Received November 01, 2019.

Sineglazov Viktor. orcid.org/0000-0002-3297-9060
Doctor of Engineering Science. Professor. Head of the Department.
Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine, (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/solar power plant.
Publications: more than 600 papers.
E-mail: svm@nau.edu.ua

Ischenko Vitaly. Postgraduate student.
Aviation Computer-Integrated Complexes Department, National Aviation University, Kyiv, Ukraine.
Education: National Aviation University, Kyiv, Ukraine (2015).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems.
Publications: 5.
E-mail: IschenkoVitaly@gmail.com

В. М. Синєглазов, В. С. Іщенко. Алгоритмічне забезпечення та програмна архітектура модуля візуальної навігації

Розглянуто проблему складності побудови програмної архітектури на бортовому обчислювачі системи візуальної навігації БПЛА. Визначено оптимальну програмну архітектуру та програмні модулі. Запропоновано реалізацію структури програмного забезпечення системи візуальної навігації та алгоритми.

Ключові слова: система візуальної навігації; згорткова нейронна мережа; комп'ютерний зір; локальні особливості зображення; детектори; дескриптори локальних особливостей; шари програмної архітектури; процеси операційної системи.

Синєглазов Віктор Михайлович. orcid.org/0000-0002-3297-9060
Доктор технічних наук. Професор. Зав. кафедри.
Кафедра авіаційних комп’ютерно-інтегрованих комплексів, Національний авіаційний університет, Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна, (1973).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки.
Кількість публікацій: більше 600 наукових робіт.
E-mail: svm@nau.edu.ua

Іщенко Віталій Сергійович. Аспірант.
Кафедра авіаційних комп’ютерно-інтегрованих комплексів, Національний авіаційний університет, Київ, Україна.
Освіта: Національний авіаційний університет, Київ, Україна (2015).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем.
Кількість публікацій: 5.
E-mail: IschenkoVitaly@gmail.com

В. М. Синеглазов, В. С. Ищенко. Алгоритмическое обеспечение и программная архитектура модуля визуальной навигации

Рассмотрена проблема сложности построение программной архитектуры на бортовом вычислителе системы визуальной навигации БПЛА. Определена оптимальная программная архитектура и программные модули. Предложена реализация структуры программного обеспечения системы визуальной навигации и алгоритмы.

Ключевые слова: система визуальной навигации; сверточная нейронная сеть; компьютерное зрение; локальные особенности изображения; детекторы; дескрипторы локальных особенностей; слои программной архитектуры; процессы операционной системы.

Синеглазов Виктор Михайлович. orcid.org/0000-0002-3297-9060
Доктор технических наук. Профессор. Зав. кафедры.
Кафедра авиационных компьютерно-интегрированных комплексов, Национальный авиационный университет, Киев, Украина.
Образование: Киевский политехнический институт, Киев, Украина (1973).
Направление научной деятельности: аэронавигация, управления воздушным движением, идентификация сложных систем, ветроэнергетические установки.
Количество публикаций: более 600 научных работ.
E-mail: svm@nau.edu.ua

Ищенко Виталий Сергеевич. Аспирант.
Кафедра авиационных компьютерно-интегрированных комплексов, Национальный авиационный университет, Киев, Украина.
Образование: Национальный авиационный университет, Киев, Украина (2015).
Направление научной деятельности: аэронавигация, управление воздушным движением, идентификация сложных систем.
Количество публикаций: 5.
E-mail: IschenkoVitaly@gmail.com