Unmmaned Aerial Image Recognition System

Student Team

Esam MashnI – Electrical Engineer
Fei Deng – Electrical Engineer
Hernan Ruiz – Computer Engineering
Jesus Amundarain – Computer Engineering
Viurniel Sanzchez – Electrical Engineering

Faculty

Dr. Ibrahim Tansel – Mechanical Engineering

Abstract:

This paper demonstrates the design procedure and implementation of an unmanned aerial system (UAS) using commercially available components and software. The UAS is a fully autonomous air system that is capable of detecting predefined targets on the ground. Communication for and between the system is achieved with the help of a 72Mhz Ch. 56 radio control, and a 3G/4G Broadband connection with the Verizon broadband network (746-787MHz, with back-up from 1900 MHz, 1.7/2.1 GHz, 700 MHz) The plane itself consists of a flat mid-wing, standard v- tail plane with a weight of 2500g to 2700g unloaded, with a MicroPilot autopilot system, two cameras, two lithium polymer batteries, and an onboard PC which will run the image recognition system and control unit. The modular system consists of three major modules: the autopilot, the image processing module, and the power systems. Our approach to target recognition is the development of a cooperative identification system. A high definition video processing card on the computer processes frames of video in order to identify the targets. A target detection filtering algorithm eliminates parts of the images that are not likely to correspond to the desired target based on expected size and solidity of the pixels. Once potential targets are identified, they are processed individually by the image recognition module that searches for shapes and letters on the potential targets.

1. Introduction

In its current form, the Miami Sky-I is an unmanned aerial system (UAS) with image recognition and multipurpose capabilities. It is built by a senior design team of students from the Electrical and Computer Engineering programs for the purpose of simulating a rescue operation mission in which the image recognition system of the plane searches for typical targets and/or conducts point reconnaissance.

The work presented in this paper will describe the algorithms that perform the target detection, the control system responsible for relaying information between software modules and Micropilot, and the different components used in the creation of the whole system.

One can define the Miami Ski-I as a fully operable desktop on a UAS. This system takes advantage of a high performance on-board computing system that is able to perform real time high definition video and image processing. The computer system is composed of a BIOstar motherboard, along with a 60 GB Solid State Drive, 8GB of RAM, an Intel i5 processor, and an HD video processing card. It runs windows 7, Matlab, TeamViewer, and Horizon from Micropilot. This system is powered by a voltage regulating circuit that provides 12V from the 14.5 onboard LIPO battery. The 12V provided are then converted into 12V, 5.5V and 3.3V connections that are needed by the computer. The aircraft accomplishes autonomous flight, landing, and takeoff with the use of Micropilot - a commercial autopilot unit. The user will specify the flight path and characteristics to Horizon (the autopilot's software) and the control unit will serve as a bridge to relay altitude and GPS information to the image processing modules in order to perform the autonomous target detection. The image processing algorithms in this design were written on Matlab and the control system module was written on C#. Matlab was chosen for image processing due to the ease of use and the vast toolbox available for image processing. Designing the control module in C# allowed us to interface easily with Micropilot through RS-232 serial communication protocols.

2. System Engineering Approach

i. *Requirement Analysis*

Considering that the main goals of the competition are the safe application and execution of Systems Engineering to develop autonomous operation in successful mission accomplishment, the Miami Sky-I followed a similar approach to its design. Basing our development on the principle of reaching key phases, tasks, and milestones, the Miami Sky-I embraces a System Engineering approach. The WBS (Work Breakdown Structure) can be divided into three main characteristics:

- Phases: These are a group of related tasks that complete a major step in the project.
- Tasks: Activities within a phase that have a beginning and end.
- Milestones: This is a reference point that marks a major event in the project.

Using this structure, Miami Sky-I was divided into three key phases, these are:

- Plane building

- Image Recognition Software Development
- Plane and Software test

ii. Design Rationale

To accomplish our final design, we used the Concept Fan approach to describe the different components and designs that could be chosen. The diagram below shows the main components that were initially considered and the final design chosen after evaluation. This concept fan is used to present the combination of the subsystems and the different modules for a final design. The UAS inherits the basic subsystems from autonomous flight systems.

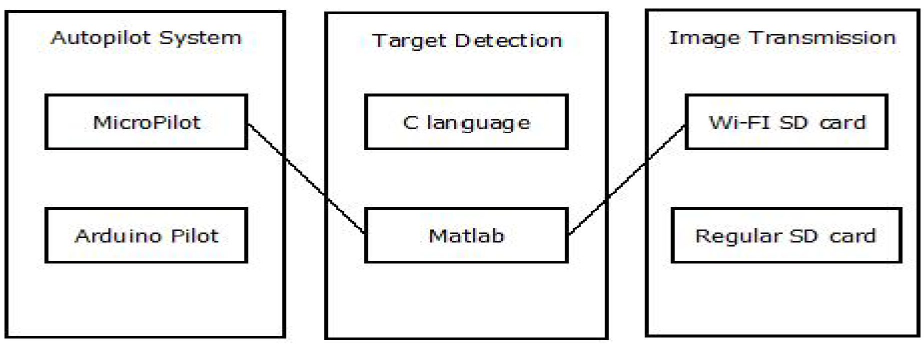


Figure 1. Concept Fan Explaining Parts of the System and their Characteristics

In the final design chosen incorporates Micropilot as the autopilot for the system, Matlab is chosen as the programming platform for target detection, and the Wi-Fi SD card (EYE-FI) is chosen for image acquisition and wireless transmission to the system. In this design, Micropilot is chosen due to availability to the component because of a donation and due to its reliable performance. Matlab provides a rich library of image processing functions that can be used to design the target detection algorithms and uses a programming language that is very user friendly. Finally, the Wi-Fi sd card allows us to automatically extract the still images acquired, which increases the ease of operation. The camera chosen for the image acquisition is a Nikon Coolpix with 16.1 MP and 14x optical zoom. Using high resolution images ensures that enough detail from high altitudes (750ft) remains of the targets in order to not only detect them but also identify the shape and letter within them without the need for zoom. On the other hand, having 14x optical zoom allows for close ups as needed but with a loss of area coverage per image. Moreover, an HD video camera with a wide field of vision (130 degrees) has been incorporated in order to detect targets located at a distance much greater than the altitude of the plane. This is ideal for low altitudes (100 – 200 ft) since it allows for a wide area to be searched with respect to the altitude of the plane in comparison with the still pictures camera that is used.

Still Picture Camera (FOV ~ 40 degrees)	HD video camera (FOV ~ 130 degrees)
Image width = 1.6 x Altitude	Image width = 4.3 x Altitude

3. Safety Considerations

Safety is of the upmost importance to the Miami Sky-I. In order to gain emergency control of the plane, a “kill switch” has been implemented that will provide emergency control of the plane to the person holding the RC transmitter. Since the range of coverage of the remote control is important in terms of maintaining control on the plane, the use of a 4G and 3G broadband network gives the Miami Sky-I the advantage of gaining a range that equals that of Verizon’s 4G and 3G coverage (Verizon is giving our project the service). In terms of stability, during the building of the plane, careful attention has been given to the structural integrity of the frame, assuring a strong and stable plane.

Before flight, all avionics of the plane are tested to ensure proper functionality. The Micropilot interface is tested by shifting the plane in all directions and observing the Horizon interface respond to these movements. A test subroutine has been written that has Micropilot move every avionic component of the plane one by one in order to ensure that autonomous control is possible. Since a hardwired connection is in place between Micropilot and the computer, the risk of communication loss between Horizon software and Micropilot is very low. Communication to Horizon from the ground is done through the broadband connection established. In the event of temporary loss of coverage or slow connection to the computer on the plane from the ground, a backup video transmission of the screen of the computer is always active that shows the Horizon interface through an RF link. If it is determined that the plane is no longer under control due to communication link through broadband connection and the RF video link, then the safety pilot acts accordingly either implementing the kill switch or flying the plane back to base if the plane is visible.

After landing, the Horizon interface is turned off to ensure that the autonomous feature of the plane does not accidentally activate and the main battery switch is turned off. Finally, in the event of a crash, all internal parts are labeled, numbered, and wrapped with bright yellow. This allows for the easy identification of parts.

4. Design

i. Hardware modules

The hardware modules on the Miami Sky-I were selected to provide a flexible platform that could be implemented on any aircraft design. This is done to provide a system that is flexible for the user and provides ease of upgrade or system reconfiguration. The hardware can be described as a system composed of four subsystems, these are: the Autopilot, the Power System, the Communications System, and the Central Control Unit (CCU).

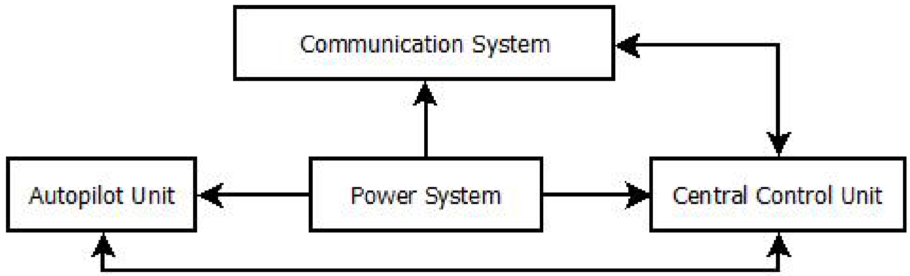


Figure 2. Visual Description of the four systems (Units).

1. AUTOPILOT

The autopilot in this system is a Micropilot. It was selected for various reasons, mainly because of its price and performance. It has the capability to perform autonomous takeoff, flight, and landing. The autopilot consists of:

- A board with a GPS , gyros, and barometer
- AGL unit with a sonar and compass module
- I/O Pins
- Serial Connection to onboard PC

The Micropilot is composed of a main board connected via serial port to the central control unit (CCU) on the aircraft. The CCU will receive information about altitude and GPS coordinates to be used in the target acquisition. The autopilot is also connected to an AGL unit, which aids in the autonomous landing of the aircraft. It is also used to determine the direction of the aircraft in relation to the north.

The I/O pins allow the autopilot to connect to the servos and ESC (Electronic Speed Controller) of the aircraft and to control these as necessary. It also allows for the safety pilot override of the system if needed. Only one operator is needed to oversee the performance of the Micropilot system. The Horizon interface allows the operator to observe the flight path of the plane, the information from the avionics of the plane, and the position of the plane. If necessary, new flight paths can be uploaded on demand by the operator through the Horizon interface.

2. COMMUNICATION SYSTEMS

The use of multiple channels of communication was implemented in order to create a system that has four levels of redundancy. This redundancy was done in order to ensure communication from/to the aircraft at all times. The main method of communication will be accomplished through the use of Verizon's 4GLTE and 3G networks through a UML290 Modem using dual 3G/4G high gain MIMO omnidirectional antennas. Although the broadband connection provides a wide range of coverage for the plane, it is susceptible to intermittent loss of signal due to altitude changes and cell hopping. In this case, the system will be backed by long range WiFi link using a high power Amplifier and two 6dB patch antennas, which complement each other in becoming a single omnidirectional antenna. The final method is a direct RF link to the plane using a 900MHz transmitter and receiver. This final link will only serve as a visual guide of the operations of the onboard computer and does not provide ground control of the aircraft but it provides information that allows the safety pilot to intervene if necessary.

3. Power Systems

The power system on any UAS plays a very important role in its efficiency and abilities. There is a tradeoff between power and weight, for this reason maximum power capabilities can be examined through the use of wind tunnels or by trial and error. The latter is the choice that was used.

i. Lithium Polimer & Nickel Cadmiun Batteries

The battery selected for the UAS's main power supply is a Lithium Polymer battery capable of holding 6000mAh of charge at 14.4 Volts. It is made with sections of highly conductive nanoparticles to improve its overall performance, and to control its discharge. This battery charges in approximately 30 minutes at a reasonable mA level. This battery is responsible for powering the motor, the onboard PC and its components, and the RF Transmitter. The NiCd batteries are used to power the Autopilot, the servos, and the RF Radio receiver.

ii. Power Distribution

A distribution board is used to convert the 14.4V input to three 12V outputs. A standard PC power supply is built to produce the 5V and 3.3V outputs needed by the PC. The power supply has been modified to fit on the plane and to remove the non-necessary outputs

4. PCENTRAL CONTROL UNIT

This system will make use of three specialized software modules. The software was built in a modular fashion to facilitate its adaptation to an environment. This allows the team to easily change or use a database or a set of databases. The major systems are: the Target Detection Module, the Shape and Letter Recognition Module, and the Control System Module. All other systems that will take mention are children to one of these mayor systems. These include camera and payload control system, the autopilot system, safety systems, and communications system.

i. TARGET DETECTION SOFTWARE MODULE

i. An Overview of the Target Detection Software Module

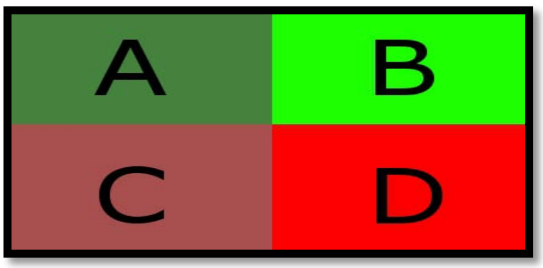
Being able to properly segment images is a very useful task in computer vision applications since it has the potential to allow for the identification of objects present in the image. Since the main purpose of this algorithm is to be able to identify potential targets on the ground from aerial images, it is very important to have effective segmentation techniques that maintain the integrity of the targets present in the image. The algorithm developed in this system performs segmentation that evaluate the frequency of color values, the amount of connected pixels, and the morphological properties of the groups of pixels. Combining these techniques with additional logic that dynamically adjusts the values of parameters based on the altitude from which the images were taken, provided very satisfactory results in the identification of targets. For testing purposes, 5 predefined targets were placed on the ground and a total of 200 images were taken during flight. After examining these images with the target detection algorithm, 22 out of 25 images containing potential targets were identified and all 5 actual targets were included in these results. Since this algorithm was designed to work in real time, special considerations were made in the streamlining of the many processes that take place as these high-resolution, 12 megapixels, images are processed. The average time taken to process each image was 2.5 seconds using a computer with an Intel i5 processor with 8Gb of ram and Matlab R2010. Furthermore, the accuracy of the overall system is greatly increased as the potential targets produced by this algorithm are passed to an image recognition software module that properly identifies the requested targets out of the potential target list.

a. Target Distinction and Identification

A constant goal throughout the development of this target detection software module was to produce a completely flexible algorithm that is able to perform as expected with aerial images of a wide variety of terrains. In order to achieve this result, it was important to create processes that produce abstractions of the image that allow further processing and successful segmentation that retains the information of the potential targets. For purposes of this algorithm, a target is described as any segmented region that is not well represented in the image. This generic description of a target means that the algorithm finds characteristics of the images that stand out like they would to the human eye.

In order to achieve effective abstractions of the image, several techniques of basic image processing and segmentation were researched. A common approach on these methods was to perform transformations from RGB (Red Green and Blue) color space and reduce the image to a black and white or gray-scale image to then perform several edge detection algorithms that can identify potential sections of interest in the images. These methods can be effective on consistent images in which the lighting, background, and other characteristics are maintained constant. Also, images where targets have a good contrast to their backgrounds are a good fit for these algorithms that greatly reduce the color information available. However, results are very poor once the composition of the images changes even in minimal ways. For example, having colors for the background and for the target that appear to be very different but have the same intensity value can yield poor results in color reduction schemes. Another important factor to consider is that, for the purposes of this algorithm, the relative size of the targets represents less than 0.1% of the overall image and their pixel value contribution to the overall histogram is very low. This low contribution means that several of the common methods used to reduce colors to black and white or gray-scale will eliminate the information of the desired target since priority is given to color values that have higher contributions to the overall histogram of the image. Other problems arise when the colors are different in hue but similar in intensity. The images below show the result from transforming a color image into gray-scale.

In figure 3, sections A and C are the same hue value as sections B and D respectively. However, the intensity on sections A and C are lower than on B and D. Figure 4 shows how the sections change after performing a color reduction to gray-scale. In this figure, sections B and D provide good contrast for image segmentation but sections A and C are very similar and would produce inconsistent results during image segmentation algorithms.



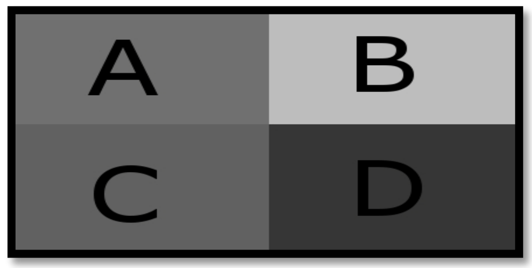


Figure 3(left), Image prior to color reduction Figure 4(right), Grayscale image after color reduction.

In order to achieve meaningful segmentation when turning color images into black and white, a proper value of threshold must be selected. Figure 5 and 6 show the effects of turning an RGB image into black and white. In figure 5, the threshold value selected was 166, and in figure 6 the threshold value selected was 167. This shows how different the results can be by using values that are only one integer interval apart. Due to the fact that the images processed by this systems are expected to be from a wide variety of terrains with changing illumination properties and other unknown details, using a black and white conversion scheme provides inconsistent results as the threshold value needs to be adjusted to maintain the target's information which is unknown at the moment of the image color transformation.

Figure 5(left), Color Image after threshold at value of 166 Figure 6(right), Image after threshold at value of 167.

b. Image Segmentation using Color Information

The color information present in the images proved to be very useful in the segmentation process. The first filter in this algorithm is a dynamic filter that adapts to the expected size of the target. Initially, the width and length of the desired targets is given to the algorithm as it is configured. Then, several calculations are performed by the algorithm in order to determine the area in pixels of the target based on the altitude from which the picture is taken and the field of vision of the camera used. This first filter turns the RGB (Red Green and Blue) image into an indexed image using a dynamic tolerance value that depends on the expected size of the target and reduces the image into a smaller abstraction. Once the expected area of the target is known, the value of colors to remove from the histogram is determined.

The filter then removes the histogram values that are well represented in the image, which in most cases represent either background colors or large objects that do not correspond to the sections of interest in the image. Since the potential targets are considered areas of the image that stand out due to their uniqueness, removing well represented color values does a good job in isolating potential targets. Figure 7 shows a sample, unprocessed, image that is fairly uniform in colors and has a target present. Figure 8 shows the result of performing the initial color and size reduction and Figure 9 shows the result of color filtering the image. Although this process degrades the detail of the target, it retains the overall structure that can be used later in the algorithm to segment the potential targets.

Figure 7, Unprocessed Image w/ target present

Figure 8, Result of color and size reduction

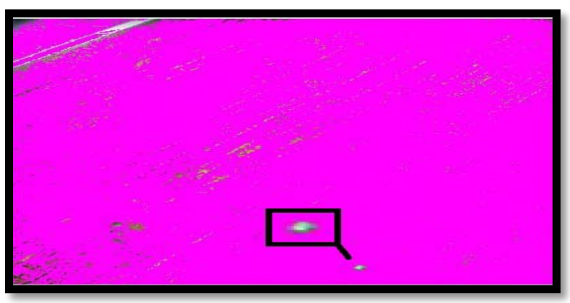


Figure 9, Result of color filtering (purple color is ignored by computer)

c. Isolating Potential Targets from the Image

After the image has been color filtered, it is then turned into a binary image in order to perform further analysis. Further filtering of the image takes place using Sobel, Canny, and pixel opening and closing algorithms. At this point the expected size of the target is known as the relationship between the pixels area and area in ft is calculated from the field of vision of the plane, altitude obtained from Micropilot, and the rage of sizes of the targets (from 2ft x 2ft to 8ft x 8ft). With this information, the potential targets can be identified using morphological properties of the connected pixels in the binary image that correspond to areas of interest. Figure 10 shows the result of performing further filtering and binary image transformation.

Figure 10, Result of filtering and binary transformation

In order to discard areas of the image that are present in the binary abstraction, four morphological properties are evaluated of the segmented sections: solidity, bounding box, area, and width to length ratio. These values are also dynamically determined from the initial calculations that produce the expected size of the target. This ensures that the changes in the apparent size of the target according to the altitude from which the images are taken are considered and the algorithm is able to adapt automatically. Figure 11 shows the final result of the algorithm that highlights the areas that were identified by the algorithm as potential targets. As it can be observed in figure 11, the target of interest is included in the results. Each of these results are cropped from the original, high quality, image and are sent to the next level of the system, the Image Recognition Module, in which further processing is performed. It is on this next level where the targets of interest emerge from these results and are identified.

Figure 11, Potential targets identified

After running the algorithm on 200 aerial test images in which 25 of them contained one of the 5 targets of interest, all 5 targets were identified as potential targets. Moreover, targets of interest were identified in 22 out of the 25 images demonstrating the high reliability of this algorithm. Figure 12 shows some of the results produced by the algorithm.

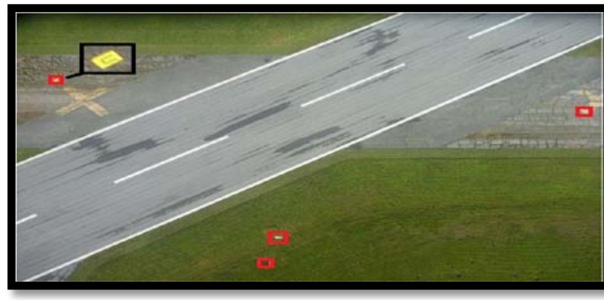


Figure 12(left). Result from Algorithm Showing potential targets on the red boxes and which include the desired (magnified in the black box) target on black. Figure 23(right) Result from Algorithm

5. SHAPE RECOGNITION AND LETTER RECOGNITION

Target recognition has become a crucible of success for the Miami Sky-I, it is also the case in virtually every mission type for the aircraft. Here we present the methods in current use by the Miami Sky-I. One major approach used for target recognition is the development of algorithm identification systems. The other major approach to target recognition is development of template matching systems. In its current form the target recognition module includes shape detection and alphanumeric recognition.

i. Metod 1 for Software Development, Shape Recognition

The 1st method is developed by MATLAB sample which calculates the area. The system estimates each object's area and perimeter. Use these results to form a simple metric indicating the roundness of an object:

$$\text{Metric} = 4 * \pi * \left(\frac{Area}{perimeter^2}\right)$$

This metric is equal to one only for a circle and it is less than one for any other shape. The discrimination process can be controlled by setting an appropriate threshold. The system uses a threshold of 0.8 so that only the pills will be classified as round. Figure 14 shows the result from this method.

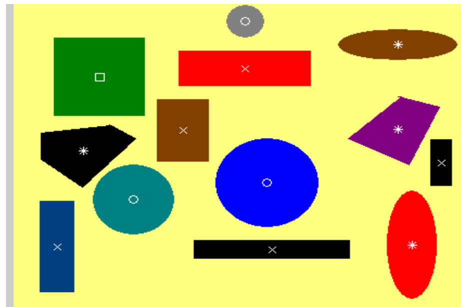


Figure 34, Shape recognition: Area method

a. Method 2 for Software Development, Shape Recognition

The 2nd method is developed in the pixels. The system arranges corner pixels which get from the MATLAB function "corners()" in a time clockwise order at first. This function can find the corners for our target. However, it is not accurate, because it will find a lot of corners for the slope line. So, after that, the system first used two closed corners to get a gradient. The system used the next corner to get the next gradient and compare these two. If the next slope is greater than the previous one more than 30 degrees or less than -30 degrees, the system defines it is a corner. After that, since there are always some fake corners at very close intervals, the system uses a threshold value for the distance between each two corner. At the same time, since the system has to arrange all the corners in the certain order in this method and the point's order is built by the closest point, it could pose problems in the footer like in the below figure. The threshold distance also can solve this problem and help one get the right answer. Below is the result, the red points are the all corners which MATLAB found. The yellow points are the real corner which the system found. The blue points are the fake corners which the system didn't choose in the program.

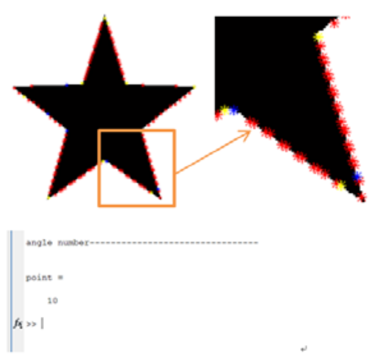


Figure 45, Shape Recognition: Pixel Method

Since this method has to see between the pixels, when the image has more noise on the edge, it can generate the wrong result. It has an outstanding limit for the real image.

b. Method 3 for Software Development, Shape Recognition

The 3rd method is trying to find out the gradient between each point with one fixed point. After that, the system was created as a matrix to put all the gradients in it. The gradients which have similar value are placed in the same line. And then, we would choose the largest value in this matrix as a line. After that, the system would flag these points and ignore them in the next circle. Below is the matrix which results from the comparison. Different lines mean different gradient. Since there are two lines that have the same length, the system will choose the first one as a line and flag it. The system was also made as a structure to save the information at the beginning of the program. This structure also can save the team number.

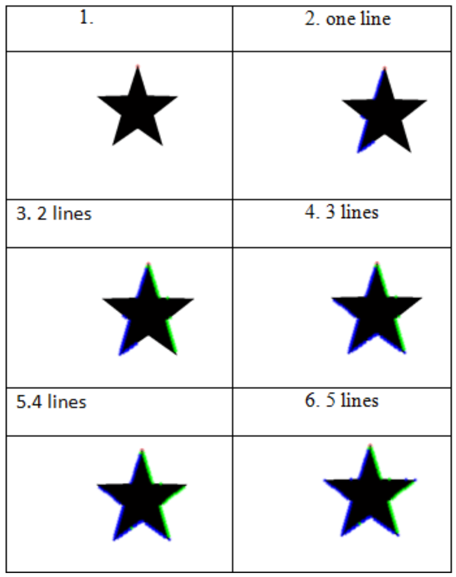


Figure 56, Shape Recognition: Line Method

After the system finds these lines, the "polyfit" function is used to get the lines. This method is a great improvement over the previous one. But it also needs to set the variables, in a similar angle range, and very carefully. In the figure 3-2, there are many mistake points in the second image.

c. Method 4 for Software Development, Shape Recognition

The 4th method is developed in Image registration which is the process of transforming different sets of data into one coordinate system. This process involves designating one image as the reference, and applying spatial transformations to the others so that they align with the reference.

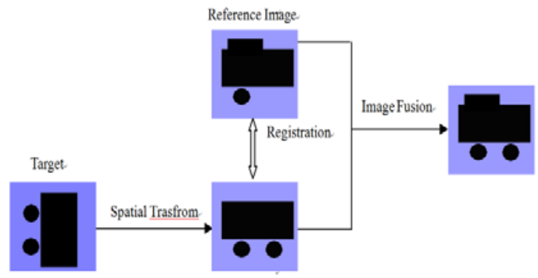


Figure 67. ***Sketch map of image registration and fusion***

The system first extracts the image feature information to form a feature space for image registration. Then the system does the spatial transformation which is determined based on the extracted features. The result after the transformation can be achieved by the definition of similarity. Then the system calculates the mutual information. So the system needs to calculate all kinds of image and calculate the MI to get the maximum value.

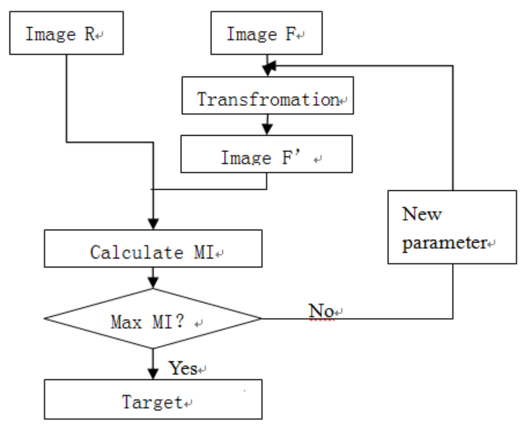


Figure 78, Flowchart of IR/Mutual Information

In our program, the system needs to make our target image have the same size with our reference image. The target image was cropped the target and resized to fit the reference. After this, the system will calculate the mutual information between the target and the reference, which is rotated one degree one time.

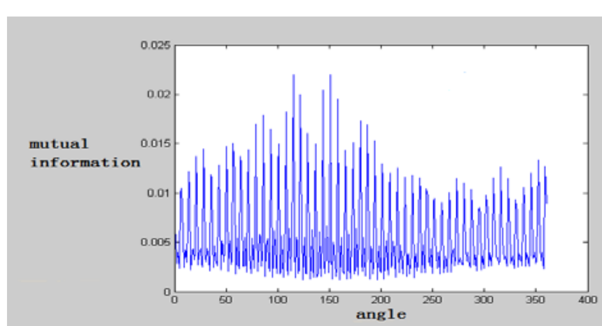


Figure 89, Relationship between the angle and Information

When the transform angle is equal to 151 degrees, the mutual information reaches maximum for 0.022. In the registration processing, based on the maximum of mutual information, the system doesn’t need the user to set feature points. The image registration algorithm depends on the image itself, and has great accuracy. However, in the registration processing, the image computing speed is relatively slow. In current research and development, this system will add an optimization algorithm to improve computation speed. The system will take nearly 10 second to finish recognition of an image.

d. Method 1 for Alphanumeric Recognition

In the alphanumeric recognition, there are two methods. The first one is the neural network. The second one is the template matching. We have develop a GUI interface to make it clear. Figure 21 shows the GUI custom interface.

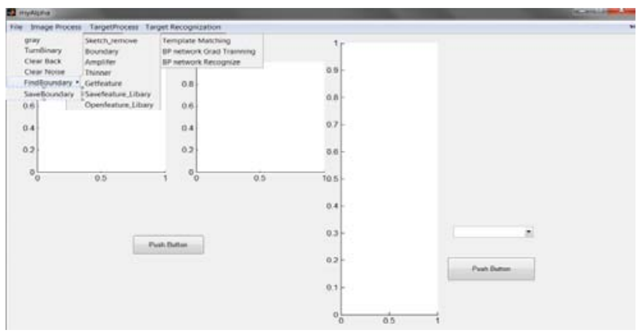


Figure 9, GUI Interface for Alphanumeric Recon

The first method is the template matching method. The program matches shapes when it recieves an image, it will clear the backgroun and clear the noise. After this, the program will change all image to 110 *110 pixels image.

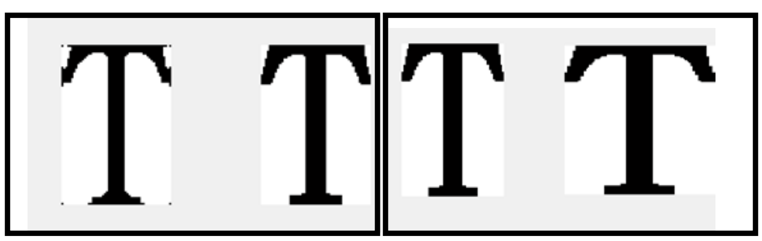


Figure 10(left), Alphanumeric Recon: 1st Step Figure 11(right), Alphanumeric Recon: 2nd Step

After this, the system can get features. Not like the shape recognition, since the letter is much more difficult to distinguish, the system is built to cut the image in the 11 times 11. The accuracy is increased after this. After we get the desired feature, we can run the template matching program.

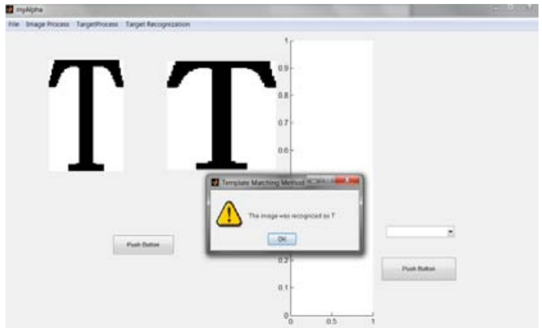


Figure 24. Alphanumeric Recon, Matching Result

The system also can show the library which contains our sample, because we built a structure to save the infromation about the letter.

Field	Value	Min	Max
num	30	30	30
feature	<121x230 doubl...	0	1
angle	<1x30 double>	0	330

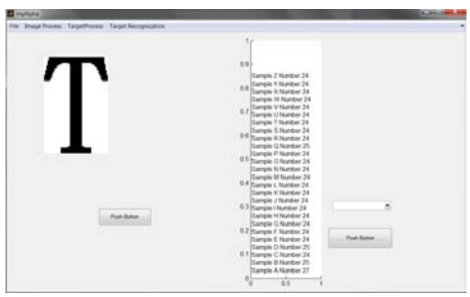


Figure 125, Alphanumeric Recon: Matching Sample Structure

If the program hasn't found the right result, the controller can save the right answer.

iii. Method 2 for Alphanumeric Recognition

The second method is neural network. This is a conventional use of autonomous target recognition using a neural network. The system should run the neural gradient training program first based on our library, like below.

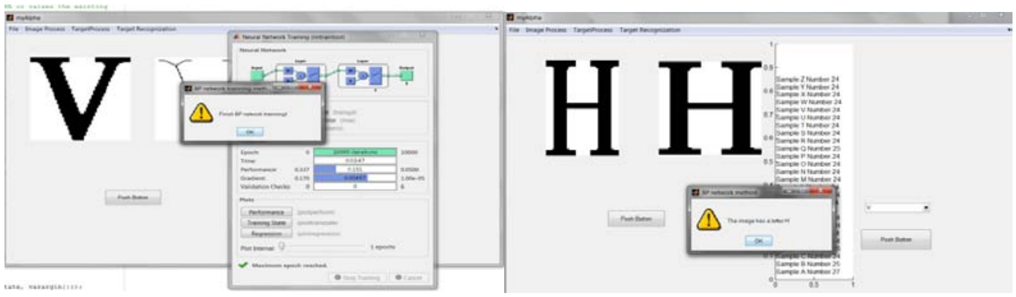


Figure 13(left),27(right), Resultant Neural Network

After these two systems are finished, the template matching methods has more accurate percentage than the neural network. There are many mistakes in the neural network, and compensation becomes very complex. So, we chose the template matching in our final program.

6. Control Unit

The control unit is capable of creating a link between different parts of the software program hierarchy with the main autopilot unit. To create the link needed for these parts, two different types of connections are built, one letting the control unit communicates with the autopilot unit, and the other connection will deal with the data transfer between the control unit and other parts of the software unit.

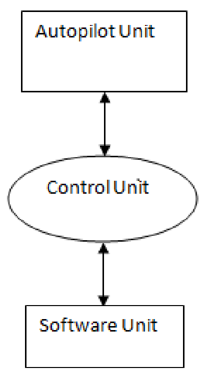


Figure 14, System Level Communication.

Before establishing a connection between the autopilot unit and the control unit, the control unit needs to set the parameter needed to open up the serial port. The main parameter needed for the serial port are: Port, Baud Rate, Parity, Stop Bit, Data Bit, Mode of communication (either hexadecimal or text), and Handshake. Once all the parameters are entered, the serial port can be open and is ready to communicate the autopilot unit until the control unit closes the connection for complete shutdown.

Information sent to the software unit consists of current position of the autopilot, the autopilot's altitude, magnetic position, east and north position, and true heading. The information the software unit needs can be taken from the autopilot unit with two special commands that can be sent to the autopilot utilizing the serial port, these commands are SSSS and GGGG. From the SSSS command, we are only interested in the following results, the altitude, and from the GGGG we are mostly interested in the HdgT, PosE, PosN, PosU, and GPS status.

Also, current commands that the control unit have on the autopilot is the ability to request and receive data, ask the autopilot unit to liftoff, and land. Control unit's current commands for the software unit are the ability to start the software unit, send data to the software unit, and terminate the process

i. Communication Between Systems

For the control unit to retrieve and send data to the other software units, the use of a Transmission Control Protocol, otherwise commonly known as TCP/IP, will enable one software program done with one language send and receive data from another software program that has been done with another programming language.

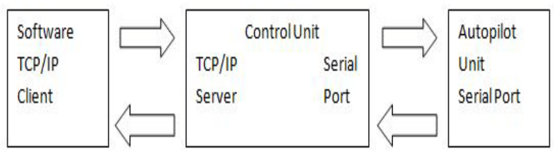
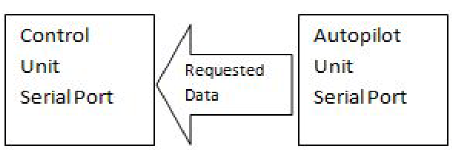


Figure 15, System Level Communication Scheme

Receiving data from other software units is done by the software units connecting to the control unit's TCP/IP server. The control unit's server creates a localhost server with a port number that the software's client knows beforehand. Once the software unit's TCP/IP client side connects to the control unit's TCP/IP server side, the client will send an encoded message that asks for the needed data to the server.

To send data, after getting the query from the software unit, the control unit will verify what type of data does the software unit need and select the correct command to send to the autopilot unit. Once the control unit knows the kind of data the software unit need, the control unit's server will call on a function that will send the needed data to the control unit's already open serial port. At the time the control unit's serial port received the query, the serial port will send the data to the autopilot unit's serial port.

When the control unit receives the information from the autopilot unit, a formatting of the, since the software unit TCP/IP client side is now waiting on the data, the TCP/IP server side of the control unit now will utilize the same connection link to send back the wanted data to the software unit, not before encoding the data with an ASCII code.



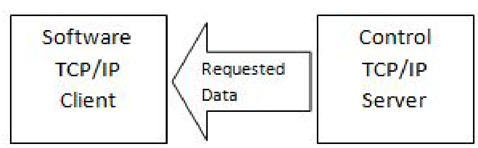


Figure 16(left), Requested data sent to the control unit. Figure 17(right), Requested data finally arriving to the software unit.

Once the software units receive the requested data, the software unit will decode the ASCII coding the control unit send. At this time the software unit's TCP/IP client connection can either disconnect or stay connected for future use, the control unit's TCP/IP server will be waiting for either a new connection to enter or read any data sent from the client side. It is important to notice that since the control unit's TCP/IP server is a running thread, other components of the control unit can continue working without any interruption from the server.

Test and Evaluation Results

- **Flight Test**

The flight capabilities of the plane have been tested by determining the performance of the plane with gradual increases of payload in order to ensure stable flight. It has been determined that the plane loaded with additional 4Lb of payload is able to safely maintain flight and computer operations for 20 minutes performing discrete increases in altitude from 100ft up to 700ft in steps of 100ft.

- **Motors Power Systems Test**

Testing of the motor and power systems depended greatly on the weight of the payload and on the altitude requirements dictated by the competition. Initial testing of the motor took place by simply securing the motor and running the computer as well as the motor on a fully charged battery until the motor ran below 30% of its capabilities – which is required to maintain control of the aircraft. This testing demonstrated that the system could run for 40 minutes on ideal lab conditions.

- **Target Detection Testing**

Testing of the Target Detection System has been performed by using the HD video camera and high resolution still images. Using HD video allows us to detect targets and identify its properties from 300 ft of altitude. Due to the wide field of vision of the camera (130 degrees), low altitudes provide us with wide area coverage. However, targets within the outer 20% of the images are hard to identify but can be detected and flagged for second fly by and still image check. Using high resolution still images, targets can be detected and identified at 700ft using minimum of 5X optical zoom. The higher the zoom level used, the better the image characteristics but the less area that can be covered by each image at high altitude. The best compromise in area coverage vs. image quality was obtained at 400ft with no optical zoom. The target detection algorithms were tested using 5 predefined targets that were placed on the ground and a total of 200 images that were taken during flight. After examining these images with the target detection algorithm, 22 out of 25 images containing targets were identified and all 5 different targets were included in these results.

- **Safety & Flight Accuracy Test**

Safety testing started on the ground. Testing of the flight capabilities of the plane, the power systems, and plane avionics was initially performed in a controlled environment. After baseline results were obtained, actual testing occurred in the air at 50% of the time duration obtained at the lab. It was determined that safe operation of the plane occurs when flight is maintained for 20 minutes. Implementation of the kill switch was never performed during flight but simulations were performed on the ground after eliminating WIFI and 3G/4G coverage, as well as RF video link. In theory, since Micropilot has hardwired communications to the onboard computer, control of the plane by Micropilot's software is very unlikely to be lost. Even if all the means of external communications with the plane fail, Micropilot will maintain control assuming no sudden loss of power or malfunction of the Micropilot unit. As the final backup of the system, if it is necessary, manual override of Micropilot is available to either implement the kill switch or maneuver the plane back using radio control.

Conclusions

The Miami Sky-I Unmanned Aerial Image Recognition System is the culmination of the efforts of an interdisciplinary team of engineering students and mentors that encompasses complex mechanical, electrical, electronics, communication, and software systems. Arduous testing and system improvements continue to be performed in order to ensure the safe and successful performance of the system. The addition of a second plane as backup is in the process of being implemented and more challenging targets are being used to test the software and camera systems.