Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 9 (2014) pp. 128-137
© MIR Labs, www.mirlabs.net/jias/index.html

Embedding ANN in UAV for Surveillance

A Case Study for Urban Areas Observation

Luiz F. Felizardo [#1], Rodrigo L. M. Mota [#2], Elcio H. Shiguemori [*3], Marcos T. Neves[#4],
Alexandre B. Ramos [#5] and Felix Mora-Camino [+6]

#Institute of Mathematics and Computing, Federal University of Itajubá
BPS Av., 1303, Pinheirinho – Itajubá, MG – Brazil

**Institute of Advanced Studies, Department of Aerospace Science and Technology*
Tamoios Highway, 1, Putim – São José dos Campos, SP – Brazil

+Ecole Nationale de l'Aviation Civile
7 Av. Edouard-Belin, CS 54005, Toulouse Cedex 4 – France

1 - lfvarginha@unifei.edu.br 2 - rodrigolmmota@gmail.com
3 - elcio@ieav.cta.br 4 - marcostheiss@gmail.com
5 - ramos@unifei.edu.br 6 - felix.mora@enac.fr

Abstract— Autonomous Unmanned Aerial Vehicles (UAVs) provide an effective alternative for surveillance in urban areas due to their cost and safety when compared to other traditional methods. The objective of this study is to report the development of a system capable of analyze digital ground images and identify potential invasion, unauthorized changes in land and deforestation in urban areas. Images are captured by a camera attached to an autonomous helicopter, flying it around the area. Two different Artificial Neural Network (ANN) techniques are used: Self Organizing Maps (SOM) to classify different photographed areas and Multi Layer Perceptron (MLP) to identify changes comparing with ancient photos.

Keywords – Pattern recognition, surveillance, UAV, autonomous helicopter, Artificial Neural Networks.

INTRODUCTION

The development of the technology of digital image processing has enabled the interest in detecting changes in multiple images of the same scene, due to the large number of applications, focusing primarily in the areas of security, soil analysis and medical diagnosis.

In the area of security, for example, there are studies that address the use of aerial robots in border areas for monitoring, [1], on the other hand, there are studies in the medical field for the detection and classification of breast tumors in pictures using ultrasound imaging features [2], and in the area of soil analysis, studies verify changes in the classification of urban areas using a multi-Layer Perceptron (MLP) [3] and Kohonen SOM [4-6]. In all these cases there is a large amount of data that must be processed so that relevant information can be extracted from the images and thus changes can be diagnosed, however, these studies utilize methods that are computationally expensive and results may take time to appear.

Also the interest has grown in the use of Unmanned Aerial Vehicles (UAVs) to collect information, initially in the military and more recently in the civil sector. Due to their low cost, there has been an increase in the need to enable UAVs into missions to collect information autonomously, in other words, without human intervention.

Currently it is very common to find areas of restricted use in many countries which need to be monitored so as to ensure their integrity by avoiding invasions, deforestation, unauthorized alterations on the ground and etc. The use of UAVs for monitoring has greatly increased in recent years, especially with the combined use of sensor data like the Global Positioning System (GPS) and the Inertial Measurement Unit (IMU) to perform the registration of the location of the images [7, 8].

Given the factors described above, the use of UAV in ground inspection of forests areas and transmission lines becomes an attractive alternative. UAVs can travel considerable distances with higher security and much lower cost than other traditional methods could provide, e.g., crewed helicopters.

With factors like fatigue and tiredness due to extensive hours of work, the human eye can often fail on the mission of detecting a change in the terrain. An autonomous helicopter is able to capture images from different angles, can be then processed by a neural network, so that the locations with the highest probability of finding invasions are spotted.

Artificial Neural Networks (ANN) is an artificial intelligence modelling technique which attempts to emulate the computational abilities of the human brain, where knowledge is maintained through the connections between neurons (synapses). The Kohonen SOM is a kind of competitive neural network with unsupervised learning [9]. For each input signal, only the neuron that has the highest affinity with it will be activated, i.e., if the inputs of the neural network are the pixels of an image, for each pixel only one neuron is activated by setting the group or class that this pixel belongs. At the end of the process, there is a new output image with pixels grouped according to similar characteristics.

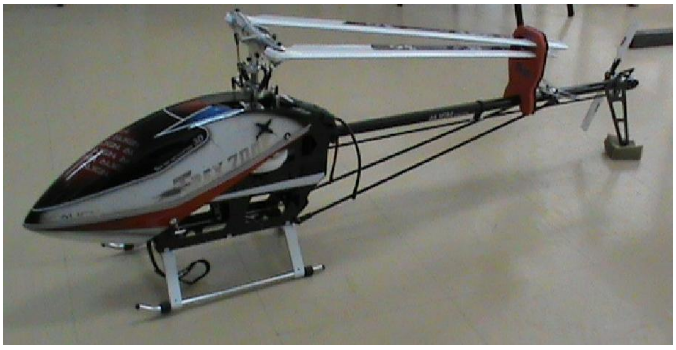


Figure 1 - Chosen Helicopter

The objective of this paper is to report a system that can collect and analyse aerial images of the ground under special areas, and elect (through images) most likely problematic places. The survey currently uses manned helicopters and it is made by human eye analyses, which has a high cost and is likely to have significant failures. Processed images are captured, in this case, by autonomous helicopters, which greatly reduces the labour and process costs.

HELICOPTER PLATFORM

Aircraft

The Mini UAV shown in Figure 1 is a commercial RC helicopter chosen for its price and performance, range, reliability, etc. It has a 1.58m diameter rotor blade, which is powered by a brushless 700MX electric, and has a maximum take-off weight of 5.2kgs.

The base RC platform weighs 3.2kgs with batteries, sensors and flight computers. With the basic configuration for GPS waypoint navigation, it has an endurance of 30 min.

A 2.4GHz RC link is used for manual flight and a 2.4GHz wireless modem is used to transmit telemetry data to the ground station and operator commands from the ground station. This link together with the omnidirectional antennas used has a line-of-sight range of approximately 5km but if by any means the link is lost, an automatic return home procedure will take command of the aircraft bringing it back to visual sight of the operator.

Helicopter Motion

The Helicopter equations of motion written in nonlinear form, according to G. D. Padfield [10], is given by:

$$\dot{x} = F\left(\mathbf{x},\mathbf{u},t\right)$$

Where F is a nonlinear function of the aircraft motion, x is the column vector of state variables, u is the vector of control variables. The state vector is:

$$x = \{u, w, q, \theta, v, p, \varphi, r, \psi\} \qquad (2)$$

where **u, v** and **w** are the translational velocities along the three orthogonal directions of the fuselage fixed axes system; *p*, *q* and *r* are the angular velocities about the *x*, *y* and *z* axes and θ , φ and ψ are the three Euler attitude angles, defining the orientation of the body axes relative to the earth.

The control vector has four components: main rotor collective, longitudinal cyclic, lateral cyclic and tail rotor collective:

$$x = \{\theta_0, \theta_{1s}, \theta_{1c}, \theta_{0T}\} \qquad (3)$$

where θ_0 main rotor collective pitch angle (rad), θ_{1s} longitudinal cyclic pitch (rad), θ_{1c} lateral cyclic pitch (rad), θ_{0T} tail rotor collective pitch angle (rad).

Starting from the following non-linearized form the force of equations of motion:

$$\begin{aligned} \dot{u} &= vr - wq + \frac{X}{M_a} - gsin\theta \\ \dot{v} &= wp - ur + \frac{Y}{M_a} + gcos\theta sin\varphi \\ \dot{w} &= uq - vp + \frac{Z}{M_a} + gcos\theta cos\varphi \end{aligned} \qquad (4)$$

where M_a is the mass of the helicopter and g is the acceleration due to gravity. The moment equations are:

$$\begin{aligned} I_{xx}\dot{p} &= \left(I_{yy} - I_{zz}\right)qr + I_{xz}\left(\dot{r} + pq\right) + L \\ I_{yy}\dot{q} &= \left(I_{zz} - I_{xx}\right)rp + I_{xz}\left(r^2 - p^2\right) + M \\ I_{zz}\dot{r} &= \left(I_{xx} - I_{yy}\right)pq + I_{xz}\left(\dot{p} + qr\right) + N \end{aligned} \qquad (5)$$

where I_{xx}, I_{yy}, I_{zz} are the fuselage moments of inertia about the reference axes, I_{xz} is the product of inertia of the helicopter about the x and z axes and L, M and N are external aerodynamic moments about the x, y and z axes. The rate of change of three Euler angles:

$$\begin{aligned}\dot{\varphi} &= \mathrm{p} + \mathrm{q}\sin\varphi\tan\theta + r\cos\varphi\tan\theta \\ \dot{\theta} &= \mathrm{q}\cos\varphi - r\sin\varphi \\ \dot{\psi} &= \mathrm{q}\sin\varphi sec\theta + r\cos\varphi sec\theta\end{aligned} \tag{6}$$

Using small perturbation theory, we assume that during disturbed motion, the helicopter behavior can be described as a perturbation from the trim, written in the form:

$$\mathbf{x}=\mathbf{x}_e+\delta\, x \tag{7}$$

A fundamental assumption of linearization is that the external forces X, Y and Z and moments L, M and N can be represented as analytic functions of the disturbed motion variables and their derivatives.

Taylor’s theorem for analytic functions then implies that if the force and moment functions (i.e., the aerodynamic loadings) and all its derivatives are known at any one point (the trim condition), then the behavior of that function anywhere in its analytic range can be estimated from an expansion of the function in a series about the known point.

The requirement that the aerodynamic and dynamic loads be analytic functions of the motion and control variables is generally valid, but features such as hysteresis and sharp discontinuities are examples of non-analytic behavior where the process will break down. Linearization amounts to neglecting all except the linear terms in the expansion.

The validity of linearization depends on the behavior of the forces at small amplitude, i.e., as the motion and control disturbances become very small, the dominant effect should be a linear one. The forces can be written in the approximate form, see equations (8, 9) at the end of this paper.

Then the linearized equations of motion for the full six degrees of freedom can be written as:

$$\dot{x}=Ax+Bu \tag{10}$$

in fully expanded form, the system and control matrices can be written as Eq. 10.1 and Eq. 12 where $a_{36}=M_p-2\mathrm{P}_e I_{xz} I_{yy}-R_e\left(I_{xx}-I_{zz}\right)I_{yy}$ and

$$a_{38}=M_r-2\mathrm{R}_e I_{xz} I_{yy}-P_e\left(I_{xx}-I_{zz}\right)I_{yy}$$

(11)

The coefficients in the A and B matrices represent the slope forces and moments at the trim point reflecting the strict definition of the stability and control derivatives. Analytic differentiation of the force and moment expressions is required to deliver the exact values of the derivatives.

All the concepts and equations presented in this section were described in detail by G. D. Padfield in his book Helicopter Flight Dynamics [10] where more detailed descriptions can be found.

Using the characteristic data of aircraft TRex700e RC it is possible to adapt the above equations to this specific model of RC helicopter and to set into the SDK of the MSFlight Simulator and using the FSUIPC.DLL to supply a feeling very (6) next to the real behavior of the aircraft in diverse flight conditions.

Autonomous Flight

All the sensors used for autonomous flight are included in a commercial integrated device, including an IMU, GPS receiver, magnetometer and barometer (see Figure 2). The device fuses the sensor data with an Extended Kalman Filter to produce a filtered state estimate of position.

The on-board flight computer uses the state estimate to generate appropriate commands for autonomous flight in the form of Pulse-Width Modulation (PWM) signals which are converted to Pulse-Position Modulation (PPM) signal through a PPM encoder board. These are fed to the servos via a Lisa/M board. This board can select between pilot RC commands or flight computer commands passing through to the servos, allowing switching between manual or autonomous stabilization and complete autonomous flight.

The current control strategy is implemented using standard techniques. Proportional Integral Derivative (PID) loops control attitude at a high update rate, while outer loops control position and velocity at a lower rate.

The PID gains have been tuned for airspeeds up to 10m/s through flight trials and system identification. This control implementation provides a reliable hover (within the error bounds of the GPS receiver), and has been used for forward flight with airspeeds of up to 18m/s. Although the controller performs well in winds with fairly constant speeds, it is not yet optimized for gusts and turbulence [11].

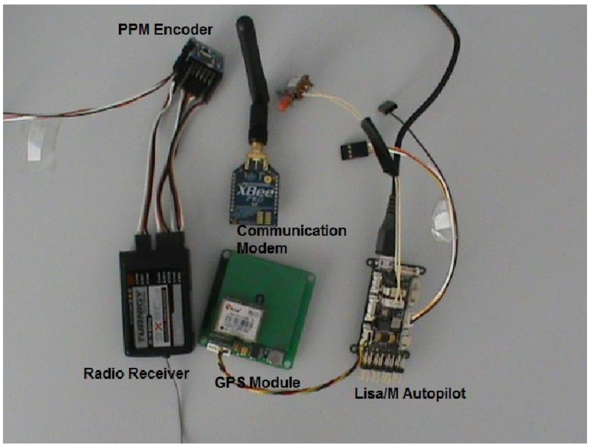


Figure 2 - Hardware Embedded in the Helicopter.

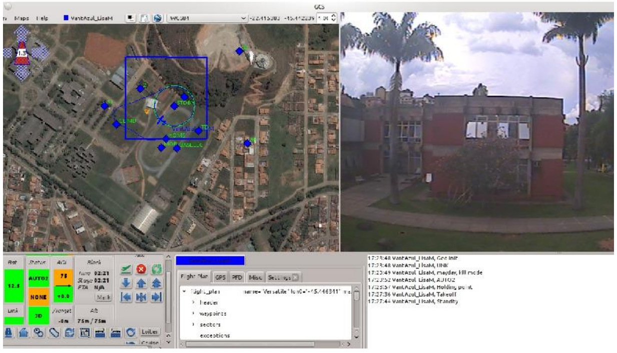
Ground Station

The software used in this project for aerial control is called Paparazzi [12]. It is a free and open-source hardware and software project intended to create an exceptionally powerful and versatile autopilot system for fixed-wing aircrafts as well as multi-copters by allowing and encouraging input from the community, shown in Figure 3.

Many changes were made in its code for supporting our helicopter. The Paparazzi project includes not only the airborne hardware and software, from voltage regulators and GPS receivers to Kalman Filtering code, but also a powerful and ever-expanding array of ground hardware and software including modems, antennas, and a highly evolved user-friendly ground control software interface.

The versatile Paparazzi Ground Control Station is a unit ground control software for micro air vehicles. It allows the operator to visualize and control a micro air vehicle during development and operation. With a flexible software architecture, it supports multiple aircraft types/autopilot projects. The purpose of the ground control station is to have a real-time monitoring of an UAV [13].

It allows the user to change in-flight parameters such as the throttle speed, altitude among others. Not only that but it also shows flight information like the battery level, current altitude, ground speed, air speed and the top view map. With a simple mouse click the user can change a waypoint location or switch to some other pre-defined route. Figure 3 shows a screenshot of the interface.



```
17:23:48 VantAzul_LisaM, Geo init
17:23:48 VantAzul_LisaM, UNK
17:23:49 VantAzul_LisaM, mayday, kill mode
17:23:52 VantAzul_LisaM, AUTO2
17:25:57 VantAzul_LisaM, Holding point
17:27:36 VantAzul_LisaM, Takeoff
17:27:44 VantAzul_LisaM, Standby
```

Figure 3 - Ground Control Station Interface.

Image Processing

A system for processing images (see Figure 4) generally has the following parts [14]:

• Image acquisition: consists of a sensor to acquire the image and a second element capable of digitizing the signal, generating a digital image.

• Pre-processing: has the function to improve the image, seeking to increase the chances of success for the following procedures. Involves techniques for contrast enhancement, noise removal and region isolation;

• Segmentation: divides an input image into parts or objects. It is one of the most difficult automatic processes in image processing. The output of this stage is usually given in the form of pixels, corresponding to the boundary of a region or its constituent points;

• Representation and description: this stage of processing the data is transformed appropriately for the computational processing that follows, stressing characteristics of interest. The process of description, also called selection of characteristics, has extracted characteristics that represent quantitative information of interest or which are fundamental in the discretion of classes and objects;

• Recognition: recognition is the process that assigns a label to an object and interpretation involves the assignment of meaning to a set of recognized objects.

The developed system classifies digital photographed images shot from a set comprising a camera[1] coupled to a support, provided with vibration absorbers the whole in turn is mounted on an electric mini-helicopter.

Images are georeferenced and the whole process of imaging is performed in a ground station that receives the photos sent by the mini-helicopter and saved in a hard disk for further processing. It allows one image to be inputted and generates a new output image where the pixels are grouped according to similar features.

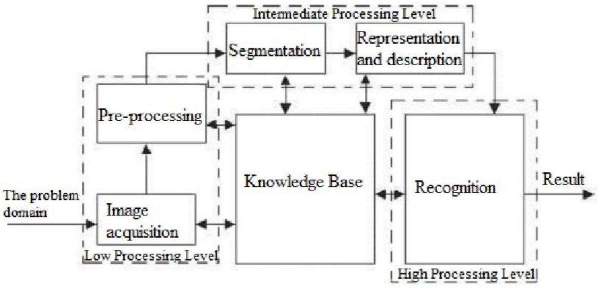


Figure 4 - Image Processing Division.

Artificial Neural Networks

An ANN is a systematic procedure of data processing based on the nervous system function in animals. It tries to reproduce the brain logical operation using a collection of neuron-like entities to perform processing of input data [15].

[1] A Sony FCB-EX980SP camera that employs a 26x optical zoom lens combined with a digital zoom function, this camera allows to zoom up to 312x.

The basic processing unit of an ANN is called perceptron, which is a primary approximation to the biological human neuron [16].

It is a decision making unit with several input connections and output links, as shown in Figure 5. A signal a_i which is delivered from an input i is multiplied on arrival by a connection weight $W_{j,i}$, so that each signal appears at the perceptron as the weighted value $w_{i,j} \cdot a_j$.

The perceptron sums the incoming signals and adds a bias b to give a total signal n. To this sum, a transfer function, usually a step-function is applied to produce the output a. Thus, if the sum of inputs reaches the threshold level, the neuron is turned "on" and a message is sent out. If the sum is below the threshold value, the neuron is quiescent and remains "off". This process is summarized in Figure 5.

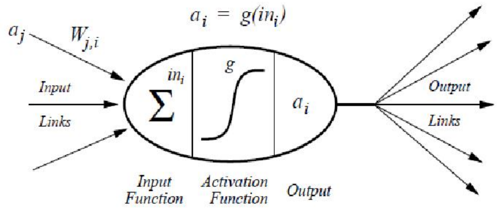


Figure 5. Schematic diagram of perceptron.

In practice, the art of ANN construction is not to use simple individual perceptron units but to link together enough of them forming multilayer perceptron neural networks in a suitable manner to solve a particular problem.

The perceptron is characterized largely by its structure, which consists of a layer of input nodes, a layer of output nodes, and one or more hidden layers. Each layer is cleanly separated, with no connections between nodes in the same layer.

The only connections are from nodes in one layer to nodes in the next layer. Typically each node in a layer is connected to every node in the next layer. Figure 6 illustrates a simple perceptron with 4 input nodes, 1 output node, and a hidden layer.

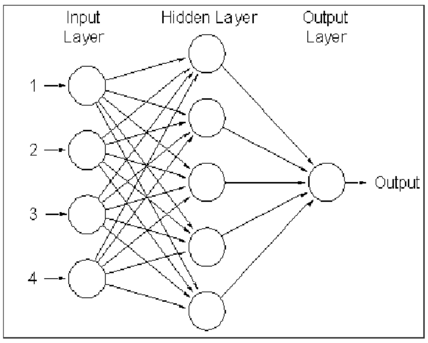


Figure 6 – Representation of a three layer perceptron (M=3).

The output layer may consist of the sum of their inputs but usually a transfer function is applied to this sum. Actually, the non-linear modelling capabilities arise because of these transfer functions. In the hidden layer, sigmoid functions are often used, whereas in the output layers, linear functions are frequently used for quantitative analysis.

ANNs are usually initialized with random connection weights then adjust the weights a supervised learning algorithm called back propagation. The algorithm works by presenting the network with a set of pre-classified training data. For each piece of data, the inputs are fed forward through the network using the current connection weights.

The back propagation algorithm then examines the output values and compares them to the expected output for this piece of data. The error in the output value is sent backwards through the network, adjusting each connection weight a small amount. This small amount is the expected value subtracted from the actual value multiplied by a small learning rate constant. Typically each item in the training set is sent through the network several times to improve the accuracy. After enough iterations, the weights will stabilize on a set of values.

The Back Propagation Algorithm

Referring to a multilayer network with M layers, N_j represents the number of neuron in jth layer, the network is presented the pth pattern of training sample with $N_0 - dimensional$ input $X_{p1}, X_{p2}, \ldots, X_{pN0}$ and $N_M - dimensional$ known output response $T_{p1}, T_{p2, \ldots}, T_{pNM}$. The actual response to the input pattern by the network is represented as $O_{p1}, O_{p2}, \ldots, O_{pNM}$, let Y_{ji} be the output from the ith neuron in a layer j for pth pattern and W_{jik} be the connection weight from kth neuron in layer (j-1) to the ith neuron in layer j and δ_{ji} be the error value associated with the ith neuron in layer j. The following is the back propagation algorithm:

1. Initialize connection weights into small random values.

2. Present the *p*th sample input vector of pattern $(X_p = X_{p1}, X_{p2}, \ldots, X_{pN0})$ and the corresponding output target $T_p = (T_{p1}, T_{p2, \ldots}, T_{pNM})$ to the network.

3. Pass the input values to the first layer, perform the output is the same input $Y_{0i} = X_{0i}$.

4. For every neuron *i* in the hidden layer and output layer, find the output from de neuron $Y_{ji} = f(\sum Y_{(j-1)k} W_{jik})$ where $f(x) = \dfrac{1}{1 + \exp(-x)}$

5. Obtain output values. For every output node I in the hidden layer or output layer, perform $O_{pi}=Y_{Mi}$.

6. Calculate error value δ_{ji} for every neuron in every layer in backward order $j=M,M-1,\ldots 2,1,$, from output to input layer, followed by weight adjustments. For the output layer, the error value is $\delta_{Mi}=Y_{Mi}(1-Y_{Mi})(T_{pi}-Y_{Mi})$ and for hidden layer $\delta_{ji}=Y_{ji}(1-Y_{ji})\sum\delta_{(j+1)k}W_{(j+1)ki}$ the weight adjustment can be done for every connection from neuron k in layer (i-1) to every layer i: $W_{jik}=W_{jik}+\beta\delta_{ji}Y_{ji}$, were β represents weight adjustment factor normalized between 0 and 1.

The actions steps 2 through 6 will be repeated for every training sample pattern *p* , and repeated for these sets until the Root Mean Square (RMS) of output errors is minimized.

Kohonen Self-Organizing Map

The Kohonen Self Organizing Map (SOM) is a map or matrix with non-symmetric dimension, where each element represents a neuron [17]. Neurons are interconnected and maintained relation to each other, even influencing each other. Each neuron or set of neurons represents an output and is responsible for a particular function. This feature is analogous to the brain where different information is controlled by different parts of the brain, such as speech, hearing and vision.

The artificial neural network resembles other brain aspects such as: knowledge acquisition from the environment through a learning process and connection strengths between neurons - the synaptic weights - which are used to store the acquired knowledge.

The goal of SOM is the classification of the input data as the knowledge that neurons acquire. The topology between input and output is always kept, since the notions of neighborhood between neurons are not altered.

The synaptic weights are initialized randomly and during learning such weights are updated at each iteration of the algorithm. The intention of this phase is to find the neuron that has the closest characteristics of the input. Thus, after a process of iteration only one neuron is activated, however its neighbors also suffer a small influence.

SOM Kohonen Algorithm

1. Randomize the map's nodes' weight vectors;

2. Grab an input vector;

3. Traverse each node in the map:

3.1 Use Euclidean distance formula to find similarity between the input vector and the map's node's weight vector.

3.2 Track the node that produces the smallest distance (this node is the Best Matching Unit, BMU);

4. Update the nodes in the neighborhood of BMU by pulling them closer to the input vector;

4.1 $W_v(t+1)=W_v(t)+\Theta(t)\alpha(t)(D(t)-W_v(t))$

5. Increase t and repeat from 2 while $t<\lambda$

Where *t* denotes current iteration, λ is the limit on time iteration, W_v is the current weight vector, *D(t)* is the target input, *α(t)* is a monotonically decreasing learning coefficient and *Θ(t)* is restraint due to distance from BMU, usually called the neighborhood function, and is learning restraint due to time [18].

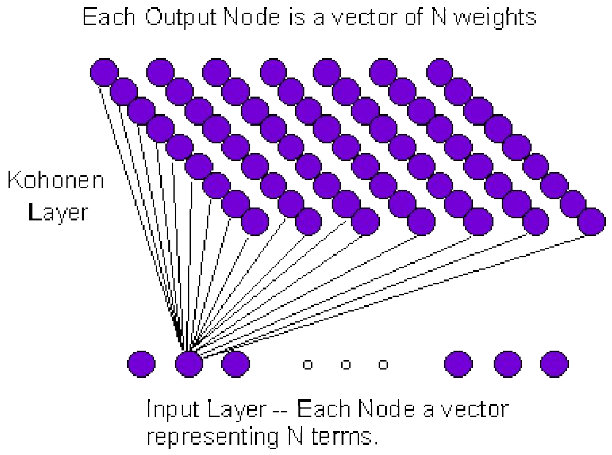

Each Output Node is a vector of N weights
Kohonen Layer
Input Layer -- Each Node a vector representing N terms.

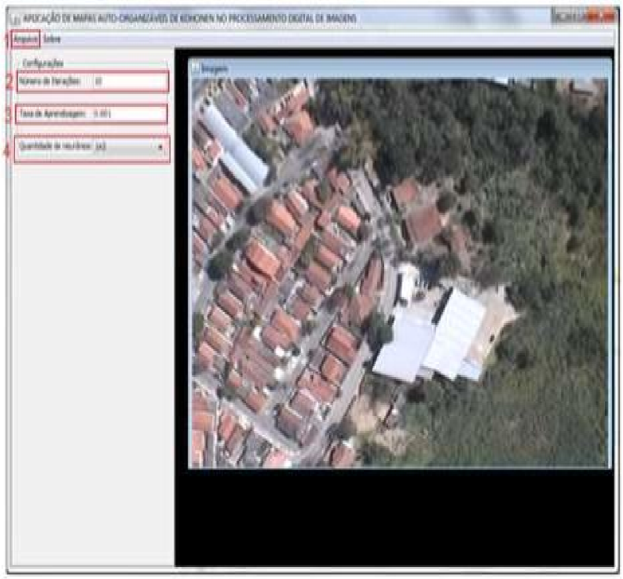
Figure 8 - Kohonen SOM Topology.

The topology of the Kohonen SOM network is shown in Figure 8. This network contains two layers of nodes - an input $+X\theta_0\Delta\theta_0+X\theta_{1s}\Delta\theta_{1s}+X\theta_{1c}\Delta\theta_{1c}+X\theta_{0T}\Delta\theta_{0T}$ layer and a mapping (output) layer in the shape of a two-dimensional grid [19].

The Software Test Interface

To the test phase it was developed a software interface that was possible to inspect interactively each photo. It was possible to set three different parameters: number of iterations, which is the number of iterations that will perform the map; and learning rate, which is a tax to define the new weight of the neuron; and the number of neurons, which determines the number of clusters, demonstrated by the number of colors in the color map.

Upon execution, it generates the color map, where you have the option to export it and / or make the reunion. Figure 9 shows the interface of the program after the selected image.


APLICAÇÃO DE MAPAS AUTO-ORGANIZÁVEIS DE KOHONEN NO PROCESSAMENTO DIGITAL DE IMAGENS
Arquivo
Sobre
Imagens
1
2
3
4

Configurações

Número de Iterações: 10

Taxa de Aprendizagem: 0.001

Quantidade de neurônios: [illegible]

Figure 9 – Test Software Interface.

With this interface it is possible to check the settings that must be made to generate color map from the image given by observing the numbered boxes highlighted:

1. After defining the attributes, the color map, can be generated;

2. Number of iterations, the algorithm of the map must perform in order to refine the final result;

3. Learning rate, which determines the new value of the weights of 28 neurons, so that it is updated;

4. Number of neurons, which determines the maximum number of groups obtained at the end of processing. To illustrate the process, a color map with the input image, with 2 groups, shown in Figure 10 was generated:

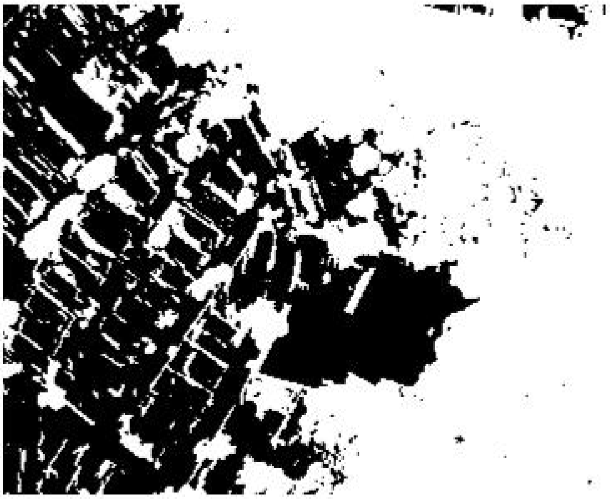


Figure 10 – Test Software Interface.

EXPERIMENTS AND RESULTS

A Java software program that implements the Kohonen SOM was developed. It allows one image to be input and generates a new output image where the pixels are grouped according to similar features.

For the processing some parameters must be set. These parameters are linked to which type of sensor is being used to capture images. For example, the optimal parameters for images captured by an UAV are not the same as satellite images. These parameters are:

- Number of iterations: how many times the algorithm will be repeated. Typically the higher the number of iterations the higher is the quality of the processing is. However, after a certain point the number of iterations starts to damage the quality of the output image. Thus, an intermediate number should be fit so that the quality is not compromised.

- Rate of learning: learning rate interferes with the updating of synaptic weights. A very small learning rate requires a higher number of iterations to converge to the desired result. However, a very high rate causes oscillations and also hampers convergence.

- Number of neurons: each neuron represents a class, i.e., a distinct characteristic. The number of neurons must be placed according to how the user wants to have some generalization in the grouping of neurons. The smaller the number of neurons, the more generalist grouping will be. A very high number of neurons will tend to cause the output to copy the entry, which is not desired.

To test the Kohonen SOM and better understand its operation some tests were conducted using aerial images of a city. Table 1 below shows the parameters used in the configuration.

TABLE
KOHONEN SOM TEST

Test #	Number of Iterations	Learning Rate	Number of Neurons
Test 0	3	0.000001	2x1
Test 1	4	0.000001	2x1
Test 2	5	0.000001	2x1
Test 3	6	0.000001	2x1
Test 4	7	0.000001	2x1

The sequences of steps taken to implement the tests were:

1. The images were converted to grayscale.
2. An adjustment in brightness and color tone of the images was made.
3. The image is processed by the software that uses a Kohonen neural network to classify different areas in the photo.

4. The Kohonen neural network output is then input in the MLP with back propagation algorithm to identify changes.

Below we can see in the Figure 11 to Figure 15, an example of test number two.

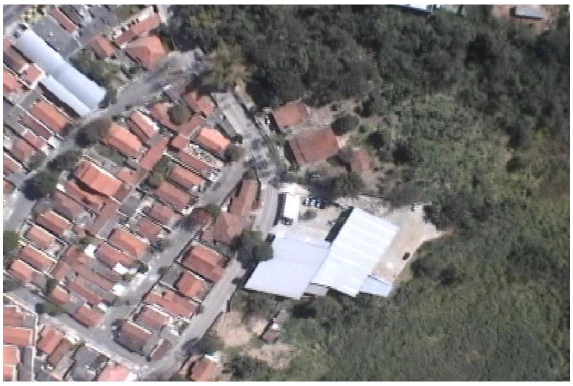


Figure 11 - Original Image Captured by the UAV.

Figure 11 was captured by the UAV over the city of São José dos Campos, Brazil.

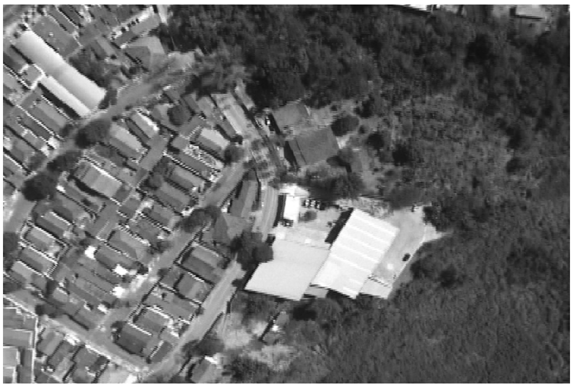


Figure 12 - Image After the Application of Step 1.

Figure 12 is a grayscale-mode from Figure 11.

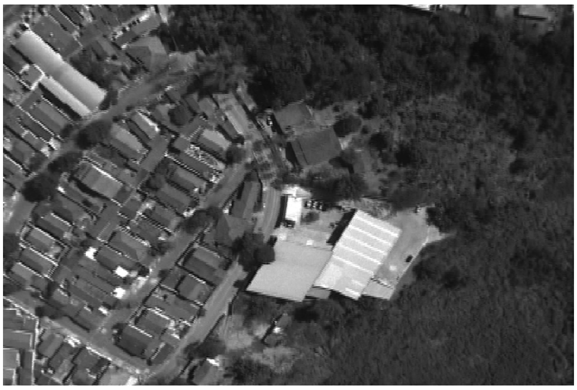


Figure 13 - Image After Brightness Change From Step 2.

Figure 13 suffered adjustments in brightness and color tone on Figure 11.

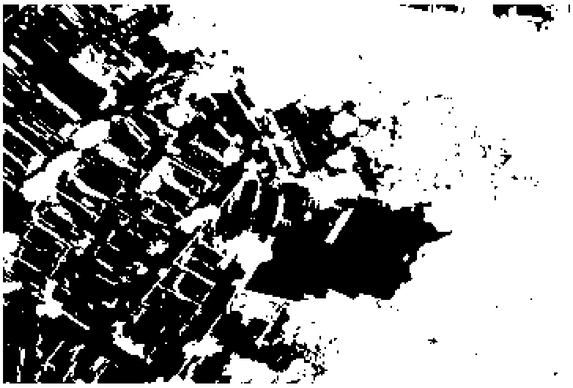


Figure 14 - Image After All 3 Steps.

Finally, Figure 14 was obtained by the processing of Figure 11 using Kohonen neural network.

The tests were made using only two neurons, one representing the color white and the other representing the color black. Steps one and two correspond to a pre-processing, which assist the image processing done by Kohonen SOM in step 3. The pre-processing makes it easier for the final process to recognize which color is closer to white (in the event trees) and witch closer to the black (the streets and the roofs of houses).

Conclusions and Future Work

Kohonen SOM allows grouping the pixels of an image with similar characteristics and MLP allows to identify the photo changes. In the case of this work, the pixels become widespread in two classes - white and black. After processing, there is a new output image with the colors rearranged. The processing result is an image capable of being quickly compared with other prior images stored in the ground station. The same process can be used to detect improper construction of houses in forests areas or transmission lines areas [20].

UAVs are already being used in many fields today and will certainly be largely used for transmission line inspection, future works include but are not limited to improvements in the autopilot system for a better stability in gusty conditions and the suitability of an airplane instead of a helicopter for other uses that require longer and faster distances for traveling.

In order to continue this project, some future works are possible, such as detecting changes in different source of images, such from satellites and other sensors. A future improvement is combining more knowledge from the images to detect better the changes. Another improvement is to classify faster the images, aiming to detect changes faster, and interpret better the results, such giving information about what changes were made.

Acknowledgment

This work has been developed with the support of institutions and companies without which its development would be impossible, therefore the authors would like to be thankful to the Brazilian agencies for research and development FAPEMIG and CNPq for the corresponding financial support the concession of the referring stock markets to the UAV Surveillance project, grant FAPEMIG 5.94/08 PA-PG (master science) and APQ 00305-11 and grant CNPq 238348/2012-1, 310031/2012-5 and 470186/2012-7.

References

[1] Gromek, A., Jenerowicz, M. Sar Imagery Change Detection Method For Land Border Monitoring, Institute of Electronic Systems , Warsaw University of Technology Space Research Centre , Polish Academy of Sciences, MultiTemp, 2011 213–216.

[2] Su, Y., Wang, Y., Jiao, J., Guo, Y., "Automatic detection and classification of breast tumors in ultrasonic images using texture and morphological features". The open medical informatics journal, vol. 5, no. Suppl 1, pp. 26–37, Jan. 2011.

[3] Del Frate, F., Schiavon, G., Solimini, C., Dept. Informatica, P. Disp, U. T. Vergata, and I.- Roma, "Application of neural networks algorithms to QuickBird imagery for classification and change detection of urban areas," vol. 00, no. C, pp. 1091–1094, 2004.

[4] Zhang, J. Kerekes, J. P. "Unsupervised urban land-cover classifications using WorldView-2 data and Self-Organizing Maps". 2011 IEEE International Geoscience and Remote Sensing Symposium, IGARSS. Vancouver, BC, Canada. July 24-29.

[5] Felizardo, L. F., Mota, R. L. M., Ramos, A. B. C., Shiguemori, E. H., Neves, M. T., Mora-Camino, F. "Using ANN and UAV for Terrain Surveillance: A Case Study for Urban Areas Observation". Thirteenth Edition of Hybrid Intelligent Systems (HIS) Conference, 2013. Tunis, Tunisia.

[6] Mota, R. "Aplicação de Mapas Auto-Organizáveis na Detecção de Mudanças no Solo ao Longo do Tempo" (in portuguese). Master Thesis. Federal University of Itajubá. 2014. Itajubá, Brazil.

[7] Y. Wang, R. R. Schultz and R. A. Fevig. "Sensor fusion method using. GPS/IMU data for fast UAV surveillance video frame registration". IEEE International Conference on Acoustics, speech and signal processing – ICASSP 2009. pp 985-988.

[8] E. Semsch, M. Jakob, D.Pavlíćek and M. Pěchouček. "Autonomous UAV surveillance in complex urban environments". International Conference on Web Intelligent Agent Technology Workshos 2009 IEEE/WIC/ACM.pp 82-85.

[9] S. Hrabar, T. Merz, and D. Frousheger. "Development of an autonomous helicopter for aerial powerline inspections", 2010 1st International Conference on Applied Robotics for the Power Industry (CARPI 2010), pp. 1–6, Oct. 2010.

[10] Padfield, G. D. "Helicopter Flight Dynamics: The Theory and application of Flying Qualities and Simulation Modelling". Blackwell Publishing Ltd. Second Edition. ISBN 978-14051-1817-0. 2007. UK.

[11] S. Haykin. Self-organizing maps. "Neural networks - A comprehensive foundation", 2nd edition. Prentice-Hall, 1999.

[12] Paparazzi Project Webpage, 2012. [Online]. Available: http://paparazzi.enac.fr/wiki/Main_Page. [Accessed: 15-Jun-2013].

[13] Paparazzi Project Webpage, Ground Control Station, 2013. [Online]. Available: http://paparazzi.enac.fr/wiki/GCS. [Accessed: 12-May-2013].

[14] R. C. Gonzales and E. R. Woods, "Digital Image Processing", 3rd edition. Prentice-Hall, 2007.

[15] Santos A.N., Soares da .W., De Queiroz, A.A.A. Low potential stable glucose detection at dendrimers modified polyaniline nanotubes. Mater. Res., v. 13, p. 5-10, 2010.

[16] GRAUPE D. Principles of artificial neural network. Chicago (EUA), Academic Press, 2007. Vol. 6.

[17] A. Machado, "Neuroanatomia Funcional" (in portuguese). Livraria Atheneu, 1980.

[18]M. Caudill, "A little knowledge is a dangerous thing". AI Expert. pp. 8(6):16–22.

[19] A. Hiotis, "Inside a self-organizing map". AI Expert. pp. 8(4):38–43.

[20] Mota, R. L. M., Shiguemori, E. H. Ramos, A. C. B. "Application of Self-organizing Maps at Change Detection in Amazon Forest". 2014 Eleventh International Conference on Information Technology. New Generations (ITNG). Las Vegas. 2014. p. 371-376.

$$X=X_e+\frac{\partial X}{\partial u}\Delta u+\frac{\partial X}{\partial w}\Delta w+\frac{\partial X}{\partial q}\Delta q+\frac{\partial X}{\partial \theta}\Delta\theta+\frac{\partial X}{\partial v}\Delta v+\frac{\partial X}{\partial p}\Delta p+\frac{\partial X}{\partial \varphi}\Delta\varphi+\frac{\partial X}{\partial r}\Delta r+\frac{\partial X}{\partial \psi}\Delta\psi\frac{+\partial X}{\partial \theta_0}\Delta\theta_0+\frac{\partial X}{\partial \theta_{1s}}\Delta\theta_{1s}+\frac{\partial X}{\partial \theta_{1c}}\Delta\theta_{1c}+\frac{\partial X}{\partial \theta_{0T}}\Delta\theta_{0T} \tag{8}$$

$$\Rightarrow X=X_e+Xu\Delta u+Xw\Delta w+Xq\Delta q+X\theta\Delta\theta+Xv\Delta v+Xp\Delta p+X\varphi\Delta\varphi+Xr\Delta r+X\psi\Delta\psi+X\theta_0\Delta\theta_0+X\theta_{1s}\Delta\theta_{1s}+X\theta_{1c}\Delta\theta_{1c}+X\theta_{0T}\Delta\theta_{0T} \tag{9}$$

$$A=\begin{bmatrix} X_u & X_w-Q_e & X_q-W_e & -g\cos\theta_e & X_v+R_e & X_p & 0 & X_r+V_e \\ Z_u+Q_e & Z_w & Z_q+U_e & -g\cos\varphi_e\sin\theta_e & Z_v-P_e & Z_p-V_e & -g\sin\varphi_e\cos\theta_e & Z_r \\ M_u & M_w & M_q & 0 & M_v & a_{38} & 0 & a_{38} \\ 0 & 0 & \cos\theta_e & 0 & 0 & 0 & -\Omega\cos\theta_e & -\sin\theta_e \\ Y_u-R_e & Y_w+P_e & Y_q & -g\sin\varphi_e\sin\theta_e & Y_v & Y_p+W_e & g\cos\varphi_e\cos\theta_e & -\sin\theta_e \\ L'_u & L'_w & L'_q+k_1P_e-k_2R_e & 0 & L'_v & L'_p+K_1Q_e & 0 & L'_r-k_2Q_e \\ 0 & 0 & \sin\varphi_e\tan\theta_e & \Omega_a\sec\theta_e & 0 & 1 & 0 & \cos\varphi_e\tan\theta_e \\ N'_u & N'_w & N'_q-k_1R_e-k_3P_e & 0 & N'_v & N'_p-k_3Q_e & 0 & N'_r-k_1Q_e \end{bmatrix} \tag{10.1}$$

$$B=\begin{bmatrix} X_{\theta_0} & X_{\theta_{1s}} & X_{\theta_{1c}} & X_{\theta_{0T}} \\ Z_{\theta_0} & Z_{\theta_{1s}} & Z_{\theta_{1c}} & Z_{\theta_{0T}} \\ M_{\theta_0} & M_{\theta_{1s}} & M_{\theta_{1c}} & M_{\theta_{0T}} \\ 0 & 0 & 0 & 0 \\ Y_{\theta_0} & Y_{\theta_{1s}} & Y_{\theta_{1c}} & Y_{\theta_{0T}} \\ L'_{\theta_0} & L'_{\theta_{1s}} & L'_{\theta_{1c}} & L'_{\theta_{0T}} \\ 0 & 0 & 0 & 0 \\ N'_{\theta_0} & N'_{\theta_{1s}} & N'_{\theta_{1c}} & N'_{\theta_{0T}} \end{bmatrix} \tag{12}$$