CHAPTER 3

3-DATA PROCESSING

As show the system architecture in the figure 2; the data and the images which came from the sensors will be send trough satellite or internet communications. SIRIO system, using VM95, can send the data at time set a priori by operators. The images collected are stored into a server, than they are processed in real time by forest fires detection algorithms.

Due to the modular architecture, the sensors can scan the whole scenario in different frequency bands and the relative algorithms can works independently to each other.

For example the images collected by thermocamera and the visible images came from the Canon powershot can be elaborated in parallel way by the correspondent algorithms producing two separated results. They can also used in cascade in order to reduce or eliminate the false alarm that the first algorithm could generate.

Hot-Spot Algorithm

The core of the hot spot identification algorithm is a radiometric model implementing a tailored progressive thresholds system applied to combination of different frequency bands (VIS, NIR, TIR) images related to the same scenario. The radiometric model evaluates the sensed scenarios by the integration of radiometric, climatologic, environmental, meteorological, orographic and vegetative characters with the sensors technical specifications. The model is based on a DEM and allows the tailoring of the identification method on the territory to be monitored. The model settings are customizable by the user, which can program the territory analysis on the basis of specific monitoring requirements and of particular characters of the area to be monitored. The model is set when the system is installed and it operates during the monitoring activities in order to be updated to the current conditions. The radiometric model evaluates the sets of multispectral images (related to the same scenario) to be processed and automatically defines the best set of identification thresholds on the basis of the overall conditions. This procedure is applied to any set of sensed images. The system, as shown in the figure 21, is thus capable in identification and localization of fire hot-spot pixels. Often, particular 'non-dangerous' features and elements (e.g. sky, houses, farms, bridges, rivers, etc) occur in sensors field of view. The radiometric behavior of these elements could affect the performances of the identification system. The algorithm provides a masking tool in order to eliminate from the images the 'non-dangerous' elements, thus reducing the false alarm rate and the computational load of the automatic procedure. The masking tool is completely programmable by the user, and masks can be added or removed in any moment.

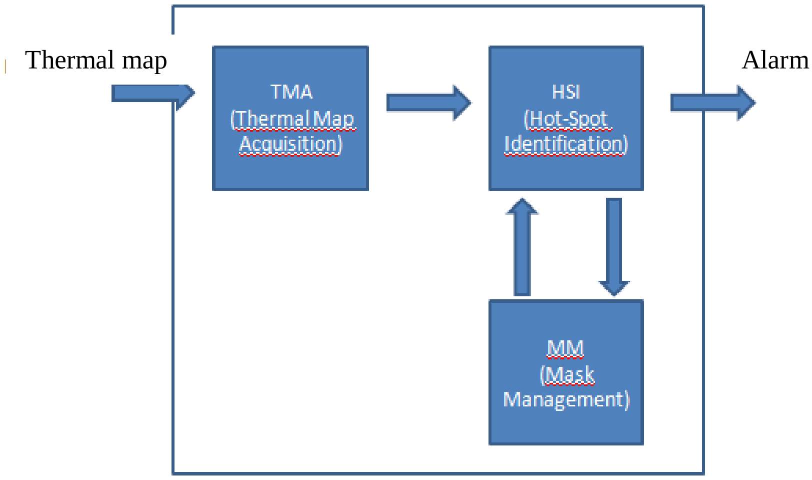


Figure 21: hot-spot algorithm architecture

The hot-spot algorithms are able to alarm if a forest fires flame appear on the scene. It is based on thermal images sensed by FLIR thermo camera.

Concerning hot-spot detection, we consider an algorithm which detect sudden increases of temperature with respect to the threshold temperature setted a priori.

Defining T_{MIN_FIRE} as the thermal threshold, T_{AMB} as the environmental temperature and DIM_{WIN} as the pixel's size processed: the lower boundary condition to consider the pixel as "event of fire" is represented by the following empiric relation:

$$T_{MEDIA} = \frac{8\,(T_{MIN_{FIRE}} - 10) + (DIM_{WIN}^2 - 8)T_{AMB} + T_{MIN\_FIRE}}{DIM_{WIN}^2} \qquad (3.1)$$

For each pixel, an arithmetic thermal average of DIM_{WIN} pixel has been considered around it. The thermal average must be bigger than (3.1) in order to recognize hot-spot pattern. 'Figure 21' show an example of thermal image in which fire is detected by the algorithm.

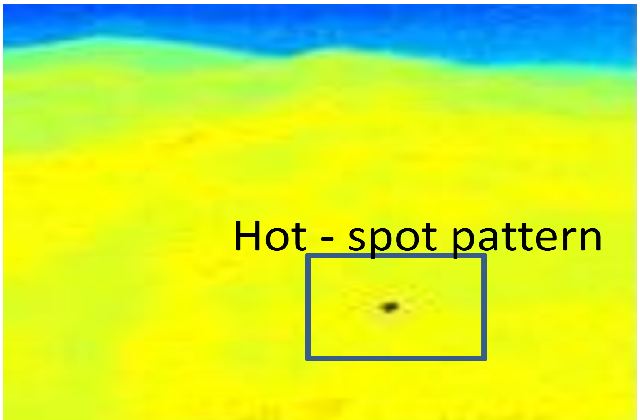


Figure 22. Example of fire detected on thermal image

The results, as mentioned before, can be elaborated in parallel or cascade by algorithms, using Visible or Near Infrared image. Visible image are taken and elaborated by Smoke detection algorithm which is able to identify a possible smoke plume on the monitored scenario.

SMOKE DETECTION ALGORITHM

In order to monitor a large portion of territory automatically, with a good cost/performances trade-off, it is necessary to develop new early warning systems. In particular, we propose an innovative method operating within a integrated surveillance system located in Italian alpine region.

Concerning early warning frameworks, the presence of smoke, and therefore its early detection, is crucial as it is now the first reminder or warning that an outbreak is about to degenerate. In many cases, the flame may not be easily seen and not detected by hot-spot detection algorithm, as for example the burning of underbrush. In order to achieve low false alarm and missed detection rate, the results are processed by smoke detection system which evaluates images in visible and/or NIR domain. In this paper, the innovative algorithm will take into account only the image taken from visible domain. As a matter of fact, this algorithm examines chromaticity changes and spatial and temporal patterns in the monitored scene. Every single pixel on images in the visible domain are composed by three component: Red (R), G (Green) and B (Blue).

Before carried out the algorithm it is necessary to say that for the detection of smoke arising among vegetation, it proposes a more data-efficient option based only on the detection of sudden increases in the B component with respect to the background. The use of the B component is owing to its greater sensitivity to the changes generated by smoke in areas in which the vegetation is predominant, when compared to the R and G components and even to the combined luminance [8].

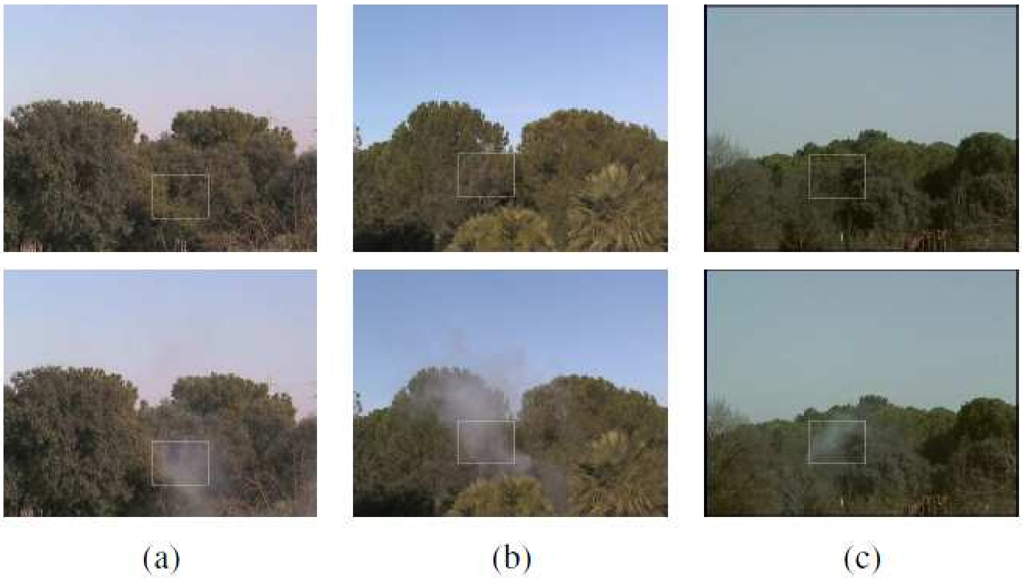


Figure 23: Three typical scenes with vegetation background without smoke and in presence of smoke

Consider the scenes in Fig. 23, here we have marked different zones where background is mainly constituted by vegetation. These frames correspond to different parts of the sequence, before and after a trace of smoke has appeared in the field of view. Table 11 shows the normalized average increase, referred to the image without smoke, undergone by each RGB component and luminance of the pixels within the marked zones in presence of smoke. It can be seen that, in all cases, the appearance of smoke among vegetation conveys a greater increase in the B component than that observed in the R and G components and the combined luminance.

	R component	G component	B component	Luminance
Scene (a)	13.8%	14.5%	20%	14.9%
Scene (b)	12.5%	13.4%	19.5%	13.8%
Scene (c)	11%	12.8%	16.4%	12.7%

Table 11: Normalized average increase, referred to the case without smoke, suffered by each component in the zones marked in Fig23

For that reason it is developed the Smoke detection algorithm which is able to recognize the B increasing between two consecutive images. In order to detect the sudden irruption of smoke in the images, the algorithm is composed by two blocks performed by two different modules.

The first block, called ‘Static’, identifies possible plumes of smoke rising from an outbreak with a scan of the chromatic pixel of captured images compared with a reference image. The second block, called ‘Dynamic’, processes the images labelled with one or more alarm pixels by the Static block output and through spatial and temporal correlations isolates effective smoke plumes from other moving features (birds, moving tree due to wind, airplanes, clouds, shadows, etc.), thus reducing false alarms that may occur at the first stage of the process. If the smoke is detected, the system sends an alarm message.

Static Block

Concerning smoke detection, we propose a method, defined in [8], and tailored for our applications, which detects sudden increases in the B component of RGB matrix, with respect to the vegetation background . The algorithm is suited for complex orography and related hard environmental conditions which could occur. The B component has greater sensitivity to the changes generated by smoke areas in which the vegetation is predominant as mentioned before in the Table 11.

The figure below, describe the Static block diagram of the algorithm; at time $t = t_0$ an image is loaded in the algorithm and waits the next image at time $t = t_0 + t_1$. At this time instant the algorithm analyzes the chromatic change between the images loaded.

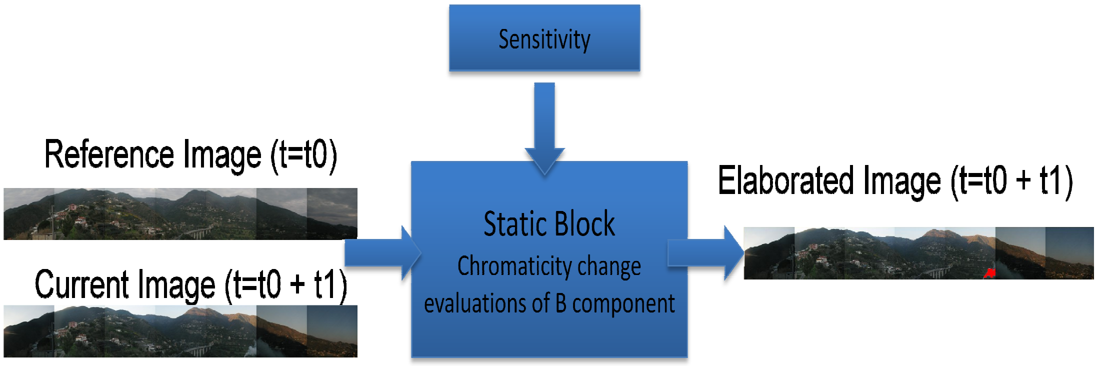


Figure 24: Static block diagram

Defining as the intensity of the B component at the bin of the reference image and as the intensity of the B component at the bin of the current image, the condition to consider the bin as "smoked bin" is :

— (3.2)

where p represent the percentage of B component set by ( ), is the maximum value of the bin in the current panoramic and is the minimum value of the bin in the reference panoramic. The equation 3.2 is fulfilled if the bin is covered by smoke. In the figure 24, the sensitivity represents the value of percentage p set a priori by operators.

It is necessary to set a p value in order to obtain the best optimal trade-off between detection and false alarm.

This operation is iterated by every single image acquisition time, the schedule depends by mechanic acquisition monitoring, dimension of scenario monitored and environmental factor.

The dimension of monitored scenario is the principal feature regarding the time schedule; for example if we monitor the same portion of scenario using zoom variable form minimum to maximum, the timing will be bigger using minimum zoom because more pictures have to be taken in order to obtain the same portion of monitored area. Afterwards, this is the main time consumption reason for the system time schedule; the choice depends to the best-optimal tradeoff between the land condition, orographic surface, meteorological condition and density area populations.

Since the system is placed into mountain on a geographical complex location, the mechanic time of image compositions has to be set in order to do not create mismatching between the neighbored picture on a panoramic overview (see chapter 2, *"Images Generated"*). Therefore the time schedule must to take into account this factor and the possible environmental factor which could cause mismatching (wind, earthquake, storm etc).

Once static block has produced the results, the output of this block is elaborated by the second step, the dynamic block which is introduced in order to reduce or eliminate the false alarm that the first block could generates.

Dynamic Block

Considering large areas to be monitored by the sensors could be very critical. Problem to face are related to objects moving too fast or too slow to respect the smoke dynamics, the luminance, the shadow of the clouds, the resolution of images etc and could generate on the 'Static Block' false alarms, false detections and missed detections. In order to work out the problem Dynamic block is been implemented. The diagram is shown on the figure below.

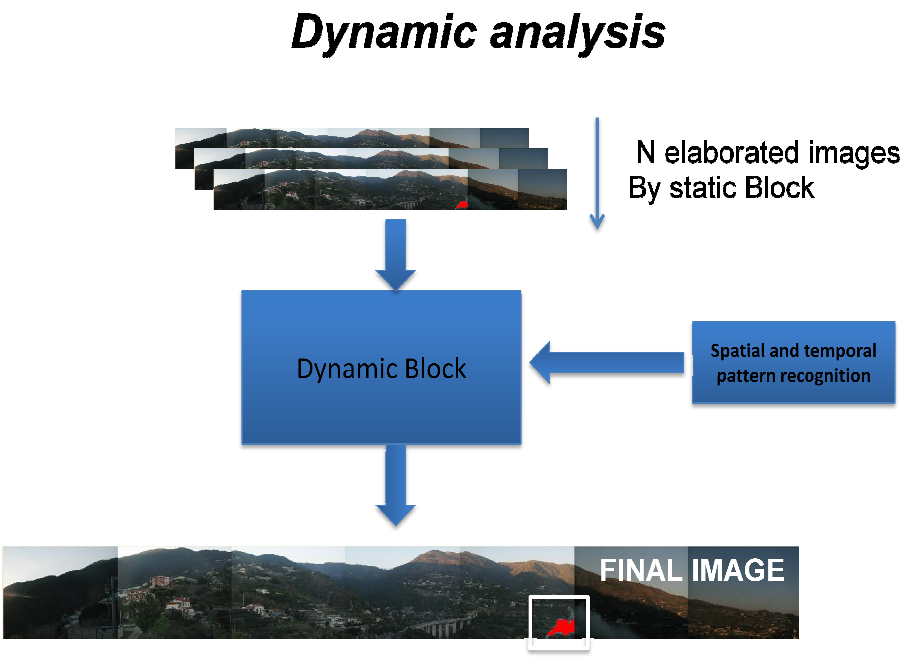


Figure 25: Dynamic block diagram

The Static block analyzes the chromatic changes between a reference image and N following images. After that. if alarms are recognized, the second step, the dynamic block, provides to analyze the bins which the first block has localized in the images as probably smoke patterns.

Therefore, dynamic block analyzes the bins just captured, through spatial and temporal correlation patterns, in order to verify if the bins considered is identified as smoke event. In other words, we are going to check out if the bins, discovered by static block, are identified as “smoked bins” or as false alarms. At the first, once bins are discovered on the current image, it is necessary to analyze if it is formed by compact regions of candidate bins.

The compact condition is:

$$Z(t)>=Zmax \tag{3.3}$$

As shown figure 26, Z(t) is the number representing how many bins are connected each other at the bin just captured, red colored in the figure below. Zmax represents the maximum value of connected bins and it is indeed proportional of the bin resolution.

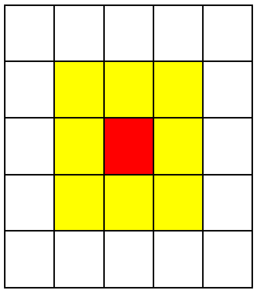


Figure 26: Bins connected

The maximum value permitted is obviously 8, because each bin has maximum 8 regions to be connected, represented in yellow. In other words, failing (3.3) means that a different and spread source is generating the changes, as for example strong wind shaking the tree top. This could reduce the number of candidate bins as smoke.

In our case, for example, we just considered a number of 3 bins connected to the bin analyzed in order to consider them as pattern alarm. this iteration is done for every single bin linked as alarm by static block. The situation could be represented as follows:

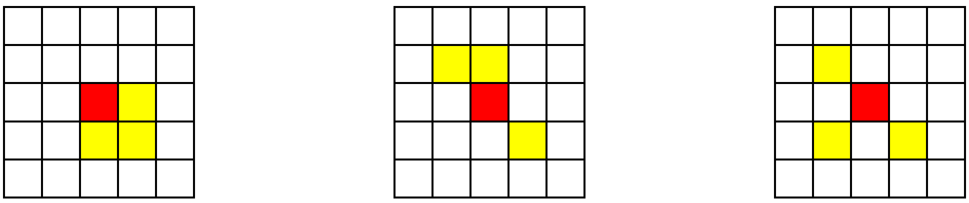


Figure 27: example of bins connected

After that, in order to consider the presence of smoke in the scene, there are several checks to reduce false alarm rate. Before that, a pattern recognition method is implemented in order to recognize the areas of interest. This is important to check out a minimum and maximum number of bins covered by smoke. Let us define Nmin as minimum number of bins in the current panoramic considerate as smoke.

N(t) represent the number of bins on the current image at the instant t. The condition:

$$N(t)>=Nmin \tag{3.4}$$

must be fulfilled. The parameter t changes every Tc, in other words Tc is a new image just captured.

In our case, for example, we set Nmin = 100, that means do not conisder pattern less 100 bins connected to each other.

In the mean time, let us define Gmax as the maximum permitted growth of smoke bin between two consecutive panoramic captured.

The condition:

$$Gmax >= G(t) \tag{3.5}$$

where:

$$G(t)=N(t)-N(t-Tc) \tag{3.6}$$

must be fulfilled in order to consider bins affected by smoke, G(t) represents the growth at the instant t between two consecutive images, t changes every Tc interval. Gmax, for example, in our case in set equal to 10000.

The values, Nmin and Gmax, are set after an initial study of possible smoke plume dynamics, this values are the best compromise between correct detection, false alarm and missed detections. In the figure below are represented the timing window related to the dynamic block decision rules.

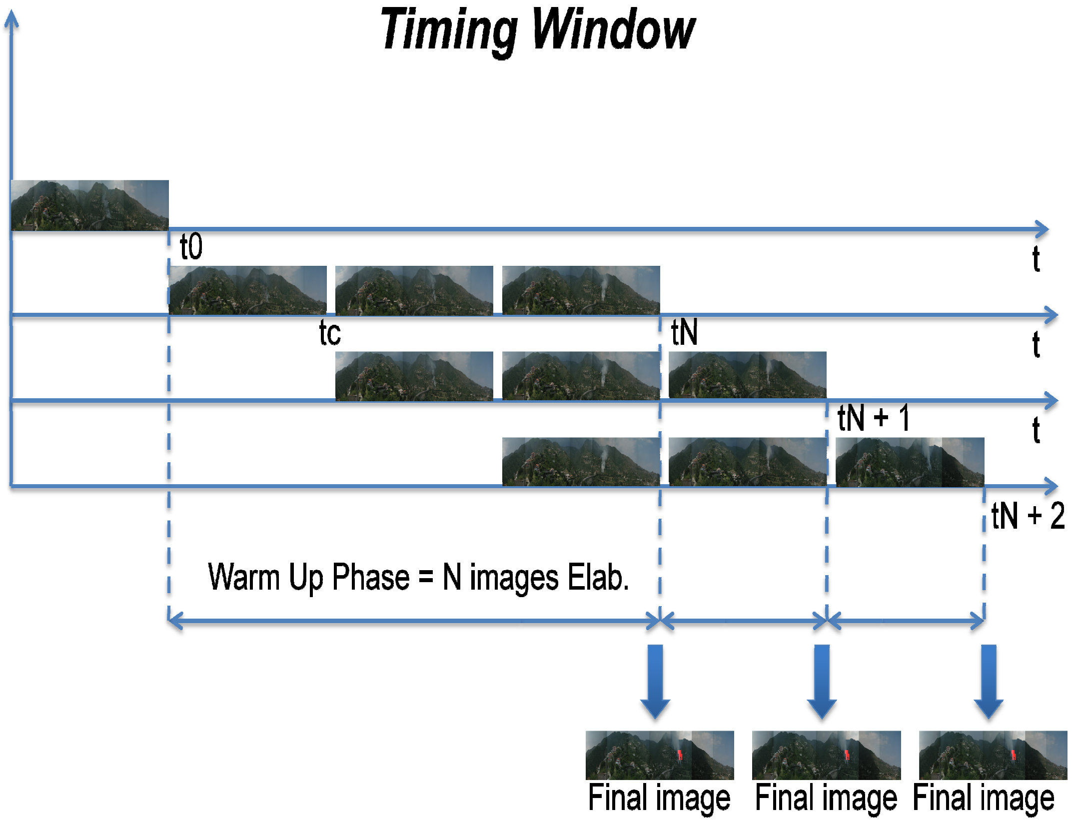


Figure 28: Timing windows Smoke detection algorithm

Every single image is taken form SIRIO at time Tc, where Tc is the time sample set a priori in order to monitor the whole scenario. At time Tc, smoke detection algorithm provide to start the analysis according to the static block phase.

At the end of acquisition and elaboration phases, the algorithms keep the status "frozen" up to new image is taken from the system. This is done for a number of time set by the parameter N, which represents the collections of images elaborated by static block. This parameter is set a priori as well. A time tN, the smoke detection algorithm decides if there are smoke plume pattern in almost 66% on the images taken and provides to start the dynamic block. The value 66% is set a priori, for example if N is equal to three, the dynamic block starts if there are alarms at least in two of three images (66%).

The values of N is set in order to obtain the best trade-off between an early warning detection and false alarm rate. If N is too big, the acquisition, elaboration and alarm detection time could introduce to much delay between the initial stage of forest fires process and intervention phase. Remember that it is important to intervene quickly in order to prevent a lot of damage caused by fires.

As shown on the figure 28, after an initial warm-up phase, where the final images come-out after N images, we can obtain a final results every time the system load a new image. This is possible because the algorithm keep in memory the last N-1 images, the oldest is deleted and the elaboration time depends on the newest image only. This iteration is done from 8am to 6pm, and it depends on the sun season because the algorithms can analyze only the visible domain. The initial time and final time could be changed, by operator, depending on the season (winter, spring, summer or autumn).

C# SMOKE DETECTION

Smoke detection algorithm is developed using C# and Visual C# tools according to image processing methods. The algorithm can operates in PC, laptop or server Windows OS based with an internet connection.

VM95 can send the images knowing the IP address of the machine used, they are stored in a directory specified. The algorithm, knowing a directory link, is able to check out, every Tc time, if almost N images are taken and sent.

For example Tc is equal to 3 minutes and N is equal to 3, in the warm up phase the algorithm give a results after 9 minutes from 8am., in this case at 8:15am; after that, it gives a results every 3 minutes(8:18am; 8:21am; 8:24am until 6 pm etc) as shown on figure 28. If an images, for some unknown reason, did not take from the system, the algorithm is able to wait until 3 images are present in the work directory. In the figure below, are represented, more in details, the results after static block, and dynamic block (Zmax, Gmax, Nmax control).

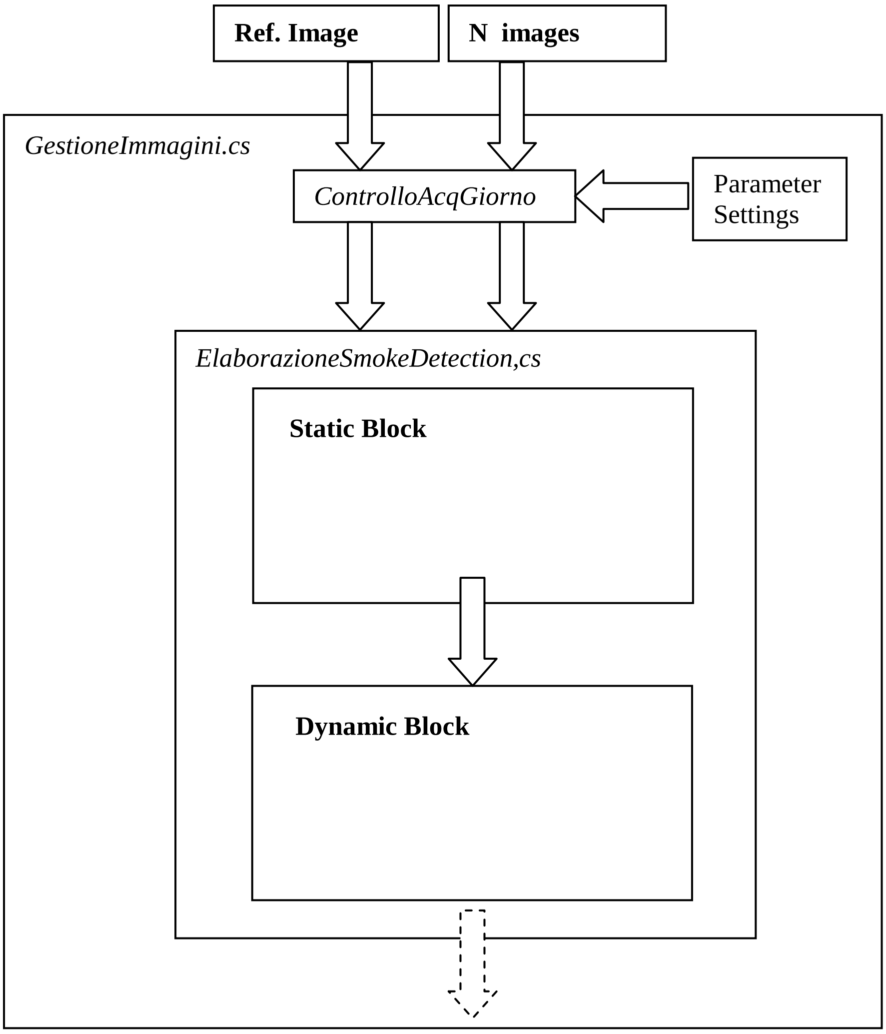


Figure 29: Block Diagram Smoke Detection

As shown on figure 29 and 30, at time a reference image is taken and is evaluated with the next N images, before that the class *ControlloAcqGiorno* defined into the class *GestioneImmagini.cs* check if the images are taken from 8am to 6 pm, defined by operator a priori on Parameter Settings file. The contol is possible because VM95 gives the name with this format: Month+Day+Hour+ Minutes+Second. In particular, the name will be shown as follow: y251435t, where y=Month (May), 25=Day, 14= hour, 35=minute and t=second.

If the ack from *ControlloAcqGiorno* is true, the static block starts the analysis according to the chromatic difference between two images as mentioned previously.

After N images the class *elaborazioneSmokeDetection*.cs starts and analyze the images.

Figure 30: images after Static analysis

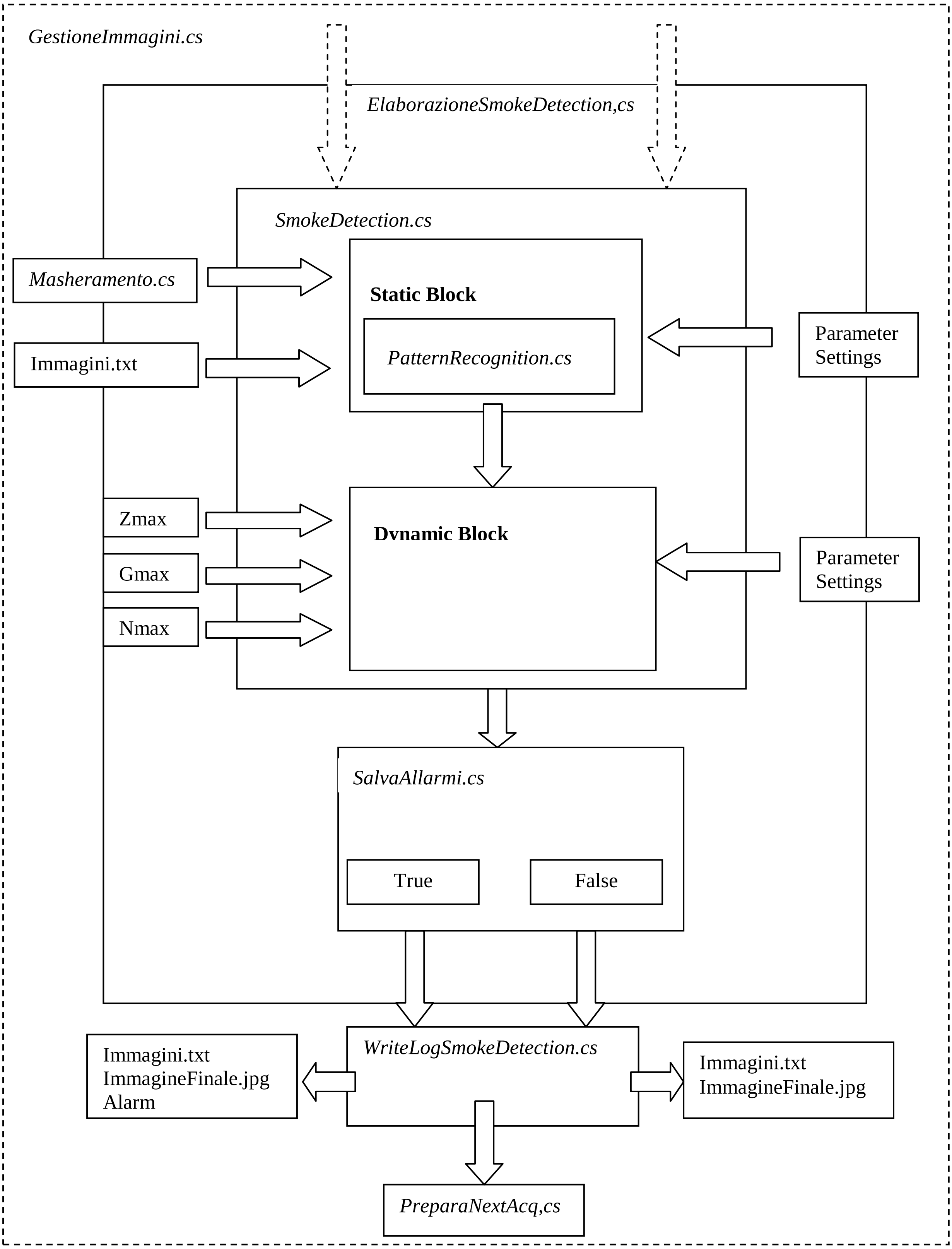


Figure 31: Block Diagram Smoke Detection Detailed

The analysis on the *elaborazioneSmokeDetection.cs* is demanded to the class *SmokeDetection.cs* after some operation of resizing and pixel grouping (Bins) with a resolution (RisX and RisY) defined a priori on the Parameter Setting file. This operations are done in order to reduce the elaboration time of the algorithm and the PC's resources. The images are elaborated by Static Block and a Parameter Setting file establish the rules of detection as: the sensitivity *p* defined in the equation 3.2 and the Pixel grouping resolution, RisX and RisY. A Mask Management methods, *Mascheramento.cs*, is able to mask the house, the sky and the features which could introduce false alarm in the elaboration phases.

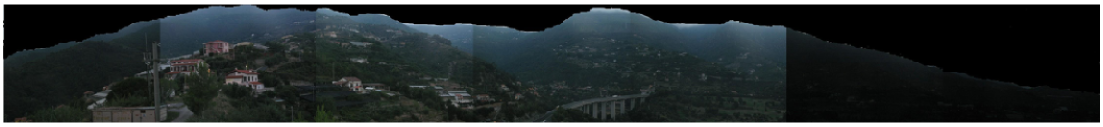


Figure 32 : Masked image

This parameters are useful in order to detect the possibly smoke plume using the methods and functions defined in the class *PatternRecognition.cs*.

If a smoke pattern is found and correctly detected, dynamic block starts the analysis using the spatial and temporal methods in order to evaluate if the pattern found belong to the smoke plume dynamics rules. This is possible, defined some parameter in a Parameter Settings file as: Gmax, Nmin, and Zmax.

Figure 33: images after Zmax, Gmax and Nmax elaborations

At time , where N is the number of images analyzed, defined in the Parameters Setting as well, the result of dynamic Block is elaborated by the methods *SalvaAllarmi.cs*. This methods is able to compose the new elaborated image (ImmagineFinale.jpg) and a file list of the analyzed images (N images), and in case of Alarm, provide to send an e-mail to the operators. The class provide also to store the images, even in case of no Alarm, on a Server directory.

Figure 34: Final image after Dynamic analysis

At the end the class *PreparaNextAcq,cs*, provides to delete the oldest image and leave the place for a new image sensed from the system. In this case we can obtain a result after only one image time acquisition.

Since the modular architecture of SIRIO system, the results can be utilized after Smoke detection analysis or can be manage by other algorithms. Algorithms developed for reducing false alarm which the previous could introduced, or algorithms for the geo-localization management and decision support tools.

In our system, we developed an algorithm which is able to link every single image pixel with a geographical coordinates pair.

GEOREFERENCING IMAGES ALGORITHM

The smoke detection algorithm provides to elaborate the images in order to detect the possibly alarm which can occur on the monitored scene. At this point we have developed an algorithm which is able to help the operator in case of fire severity. It provides to give some important help instruments in term of additional layers support for the extinguish operations focusing on the forest fires event with a precision spatial location. The spatial location is given by the forest fires location calculated trough geometrical and projective transformation on the image elaborated. This transformation are able to link every pixel, or bin, of the images, with a latitude and longitude coordinates, in particular those affected by fire or smoke. Since the forest fires coordinates are found, the algorithms automatically provides to detect some important additional support layer as: water point supply, access-way, helicopter landing and operation squad locations. In particular, highlight those are close to the fire severity.

The algorithm, as mentioned before, is build up according image geometrical and projective transformation using the Digital Elevation Model (DEM) with a resolution of 100 m.

DEM

A digital elevation model is a digital model or 3-D representation of a terrain’s surface. It represents the bare ground surface without any objects like plants and buildings. DEM can represented as a raster (a grid of squares, also known as a heightmap when representing elevation) or as a vector-based. he data are either collected by a private party or purchased from an organization such as the U.S. Geological Survey (USGS) that has already undertaken the exploration of the area. Digital elevation models are gray scale images wherein the pixel values are actually elevation numbers. The pixels are also coordinated to world space (longitude and latitude), and each pixel represents some variable amount of that space (foot, meter, mile, etc.) depending on the purpose of the model and land area involved. In our case every single DEM pixel are georefered as UTM or DEG. In the figure below is shown an example of DEM used.

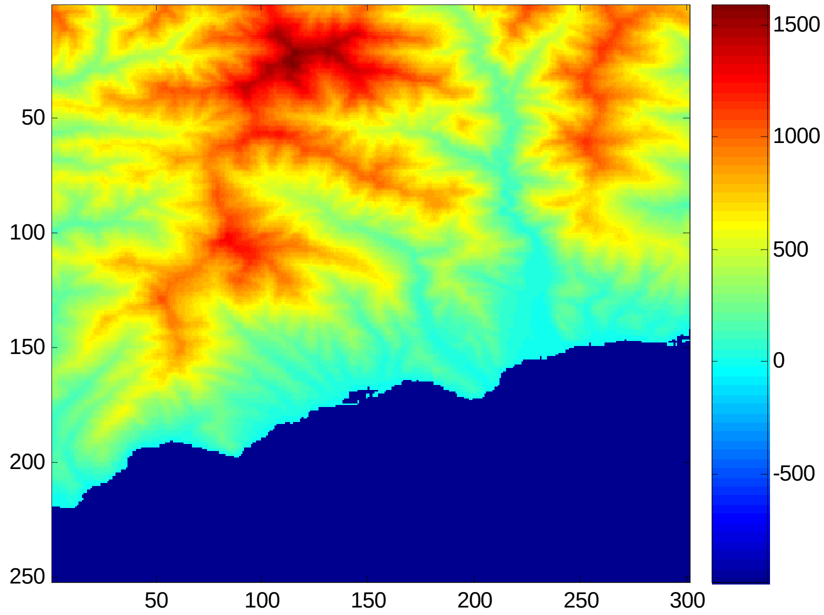


Figure 34a: Digital Elevation Model Example (2-D)

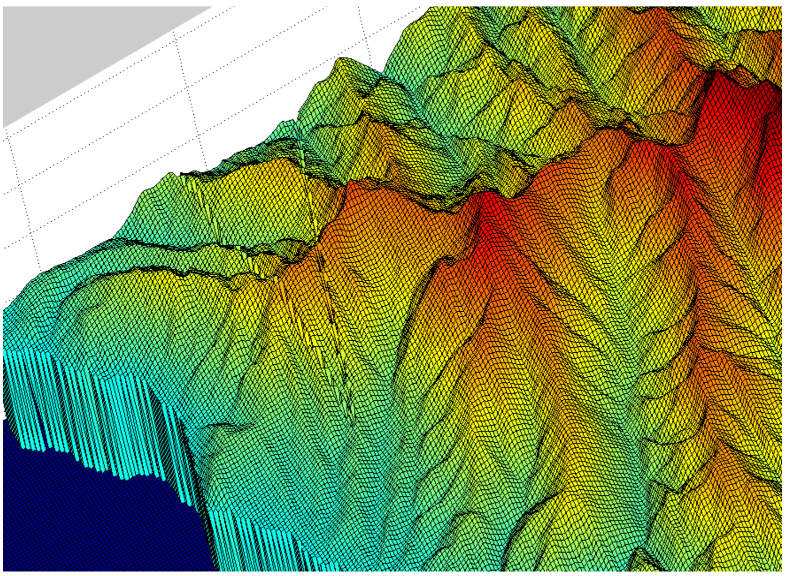


Figure 34b: Digital Elevation Model Example (3-D)

The DEM is a useful way to build up the Field Of View (FOV) of camera. The FOV, is a cone of view and represents the scenario which is monitored by the system, it depends by several sensor factors as: sensors size, zoom, focal length etc. there are other important factors which are taken into account in order to build up it on the DEM: the vertical and horizontal inclination, the North orientation, the system position and the horizon elevation.

FIELD OF VIEW

The field of view is defined in chapter 2. The field of view depends on the zoom, focal length and sensor size parameters. As calculated in 2.4, the Vertical field of view is 20.3° considered the focal length with a horizontal field of view equal to 27°. At this point we are able to overlap the total FOV into the DEM using geometrical and projective transformation.

The north orientation is given by a compass and it is calculated for each position of the camera, as shown on table 6. The horizon elevation and geographical position are given by GPS receiver. and All this factors are useful and necessary to build up the FOV. The figure below show us the field of view calculated for scene number 1.

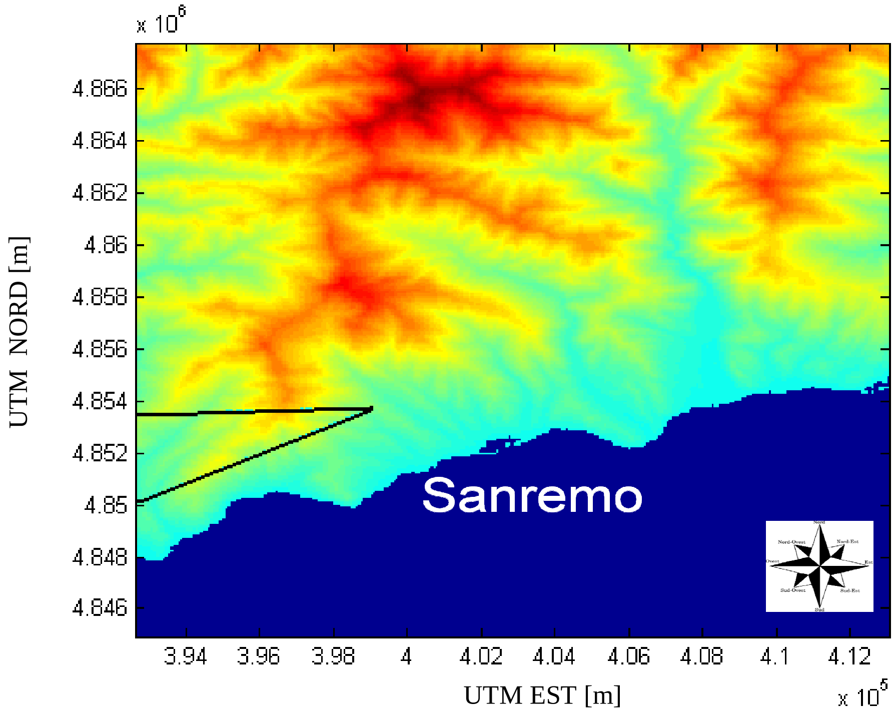


Figure 35: FOV on DEM

It is automatically generated and it is possible to obtain information regarding altitude and geographical coordinates for each pixel inside the field of view. This view corresponds to the image shown on figure 36.

Figure 36: Scene number 1 taken from SIRIO

GEOREFERENCING TOOL

Once the field of view is calculated, we can link every single image pixel with the geographical coordinates, both in UTM and Degree [19]. The way of procedure is defined below.

First of all, vertical and horizontal image resolution must to be knew because the field of view on DEM will be subdivided into portion equal to these information, the figure below describe the vertical subdividing.

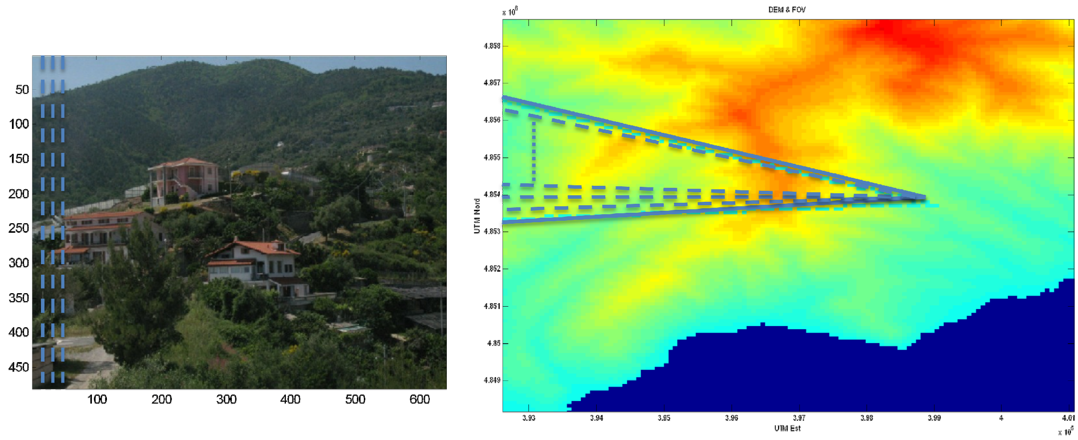


Figure 37: Geo-ref. tool horizontal subdividing

Each DEM sub-portion corresponds to each column of the image considered. The first portion is taken into account and a profile will be extracted. Figure 38 show the profile extracted, on the x axis there is the UTM EST (Longitude) value, while in the y-axis there is the altitude in meters. The start position depends from the earth surface and the building altitude.

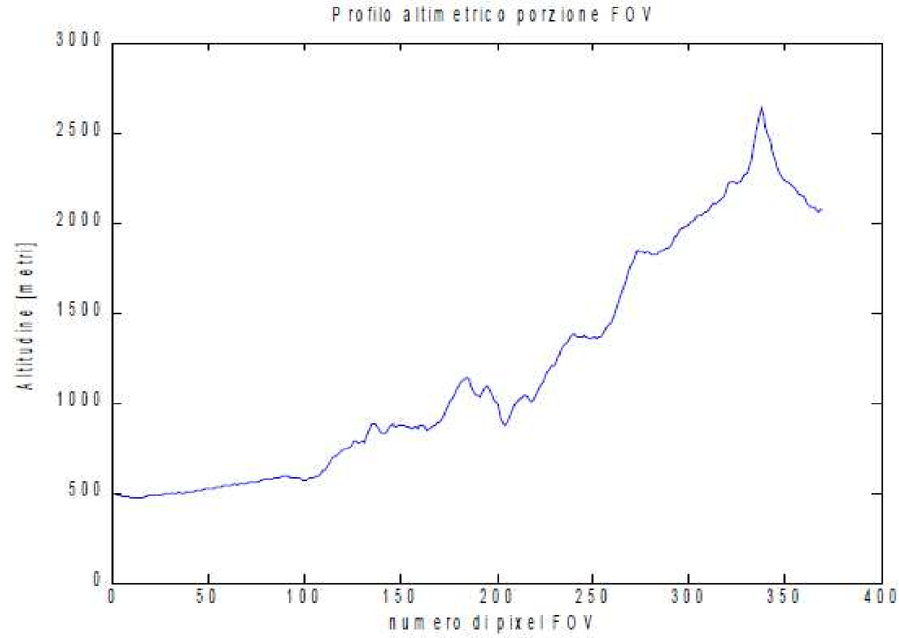


Figure 38: Profile extracted form first horizontal portion

The profile will be sub-divided into portion equal to the horizontal image resolution, each profile sub-portion corresponds to each row of the image considered. As shown on figure 39, the dotted lines scan the field of view form bottom to top.

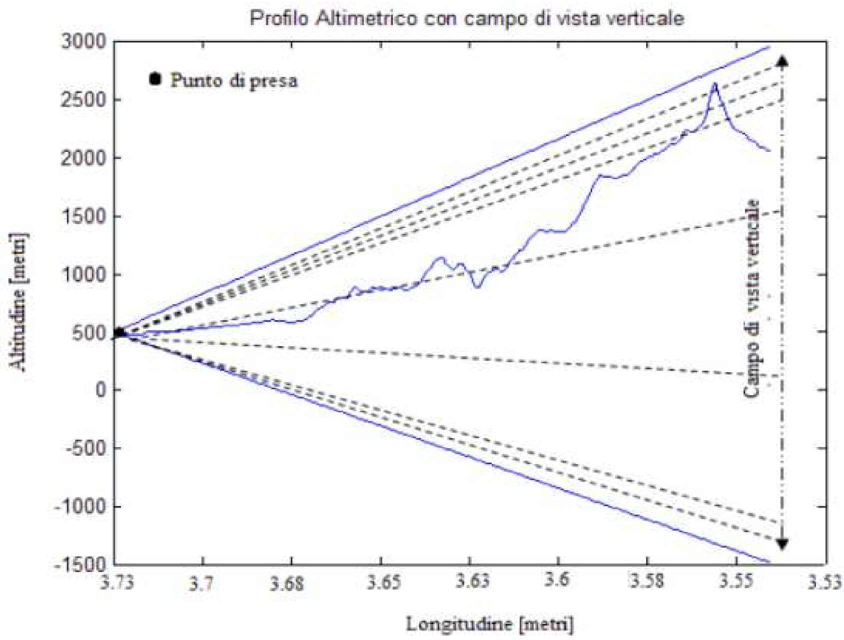


Figure 39: Portion with dotted lines

When the dotted line across the profile at least in one point, the longitude information will be take into account. For example, the first dotted lines correspond to the first pixel and this pixel has linked with the longitude information is retrieved as shown on figure 40.

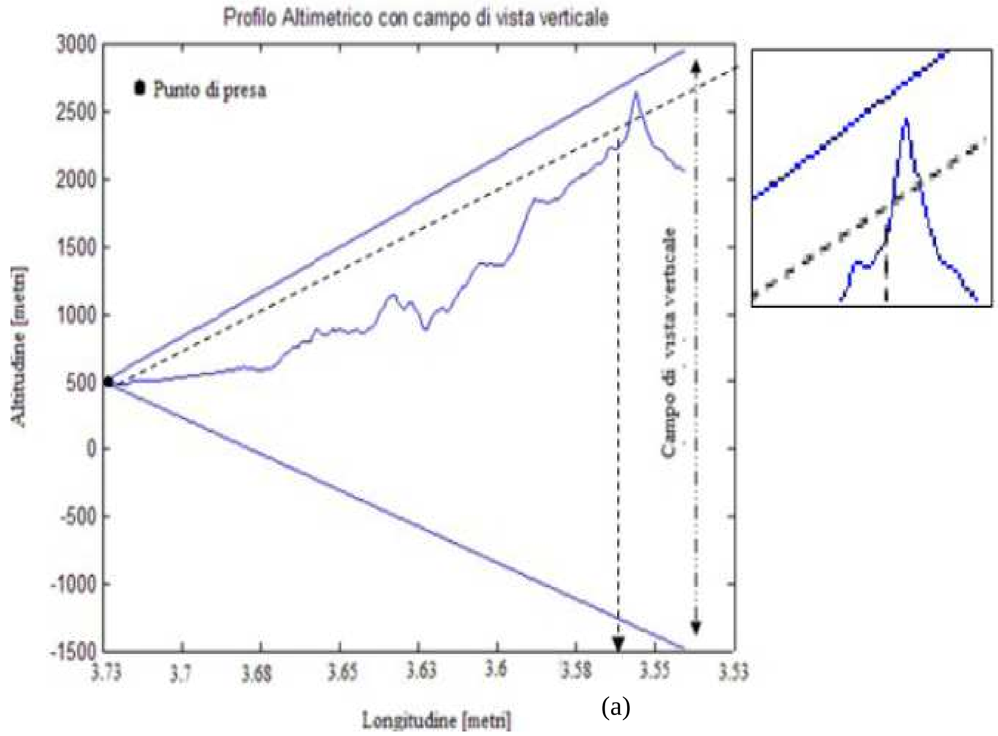


Figure 40: Longitude found

Using DEM and considering the previous horizontal field of view portion, the longitude value found, point (a), is associated to the relative latitude point (b).

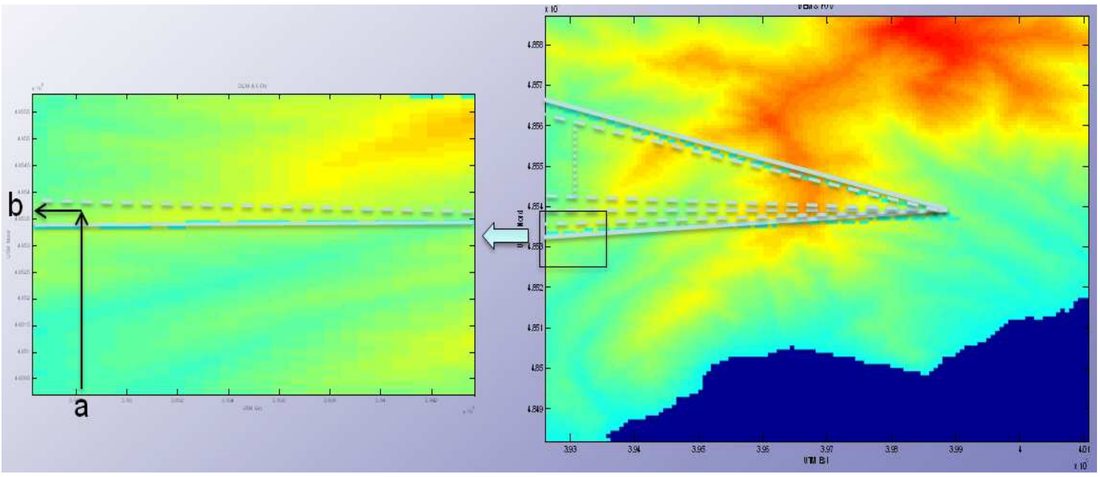


Figure 41: Longitude and Latitude found

This operation will be iterated a number of times equal to the vertical and horizontal image resolution. The results are linked into a matrix NxM, where N is the horizontal image resolution and M is the vertical image resolution.

The operation is automatically and needs only some input at the beginning, no ground control point are taken a priori to geo-reference the images. The algorithm could work for just one time in order to generate a matrix with the geographical position. This file is elaborated by a decision support algorithms which analyze the results from smoke detection and geo-referencing tool. Decision support methods also needs additional information layers which are overlap on DEM, or on Google Maps, in order to obtain which are more close to the fire event.

MATLAB® Georeferencig tool

The algorithm for image geo-referencing is developed in Matlab®. It is automatically generated, needs only some feature for beginning. The features are: latitude and longitude camera position, north orientation, inclination and horizon elevation for each scene (see table 6).

In the diagram below it is shown the blocks which constituted the Geo-referencing tool.

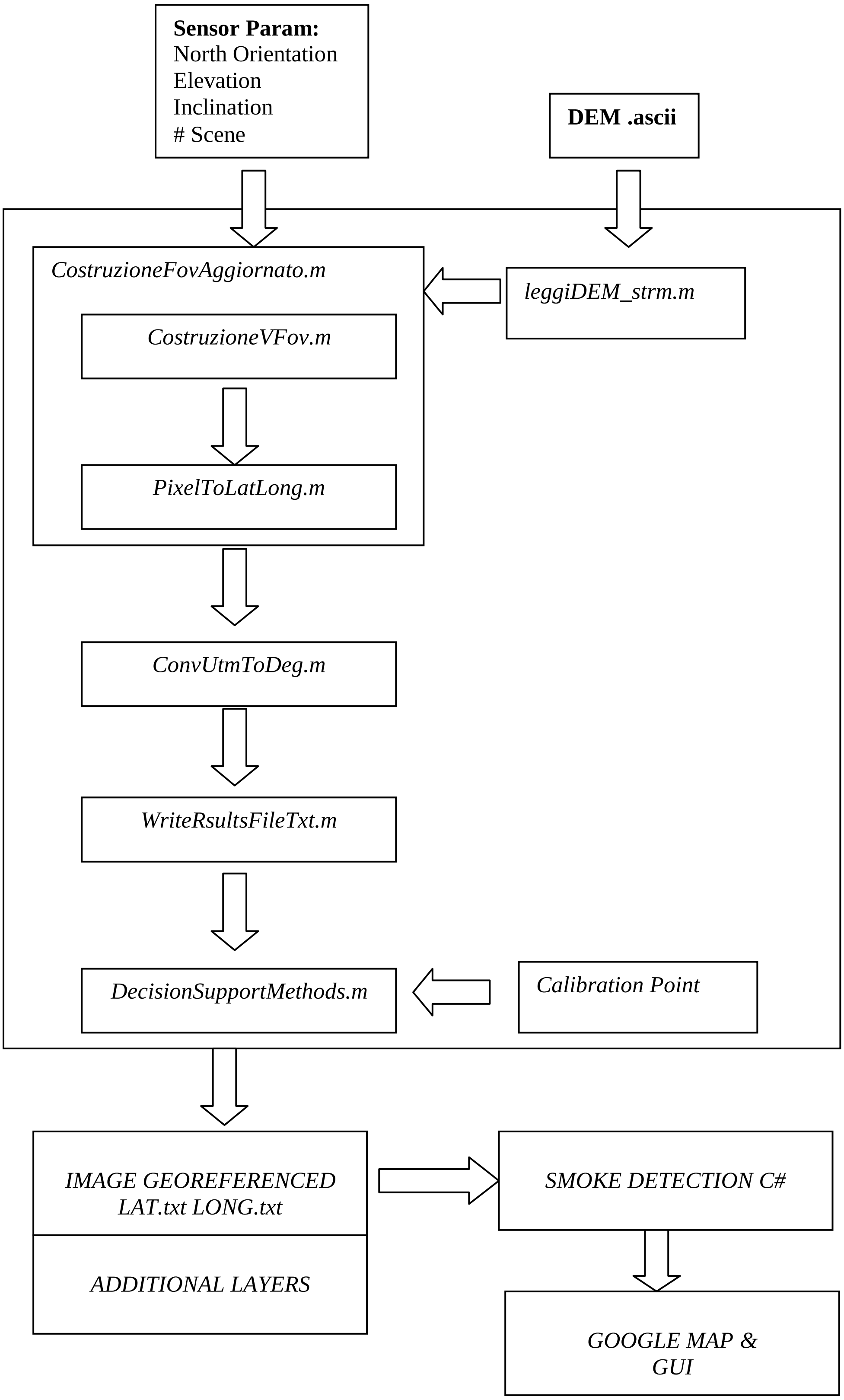


Figure 42: Georeferencing tool blocks dyagram

The DEM free downloaded from strm web site, is a large portion of territory with a spatial resolution about 100 meters in both x-axis and y-axis. It was available in ascii and a small function *leggiDEM_strm.m* has been developed in order to generate a matrix, with DEM's size, which, for every single cell, contains altitude and geographical information. The large part of territory, as shown on figures 43a, 43b and 34c. It was resized for time elaborations reasons, focusing on the territory which is included the monitored scenario.

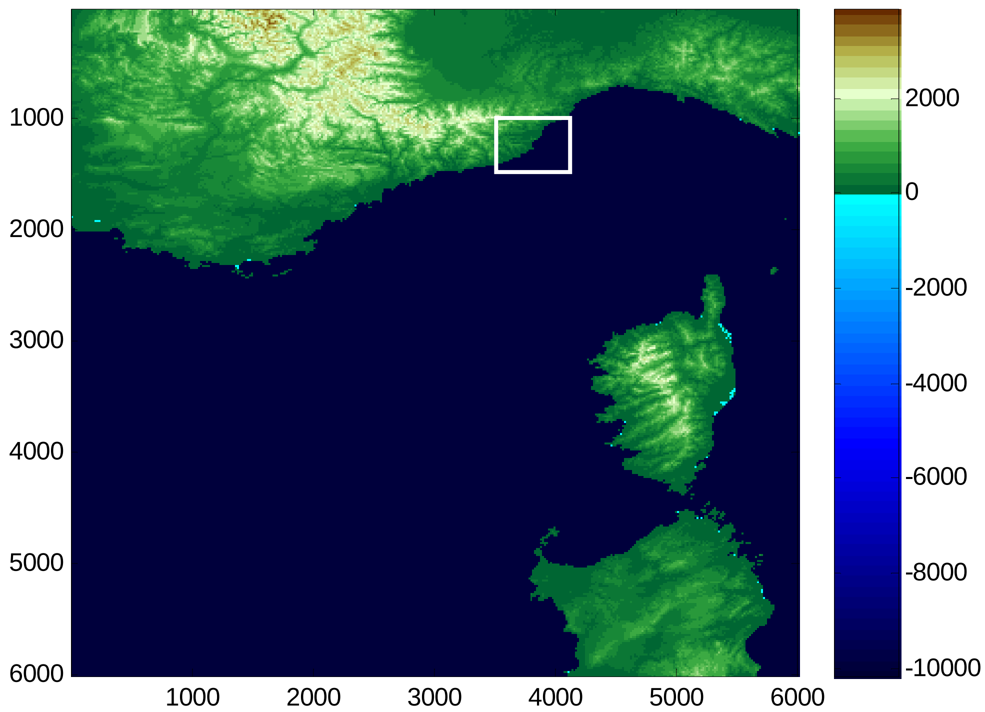


Figure 43a: DEM from strm after *leggiDEM_strm.m*

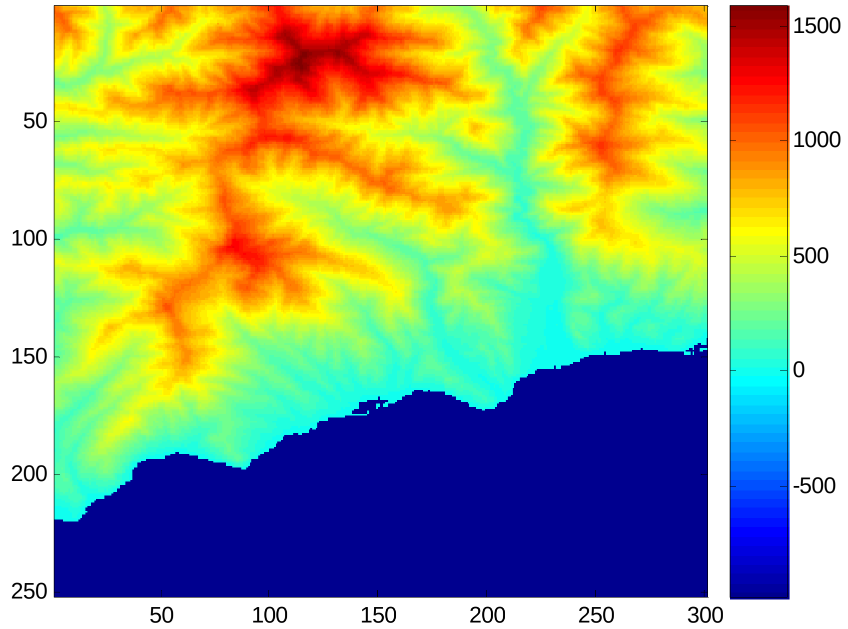


Figure 43b: DEM resized from strm after *leggiDEM_strm.m*

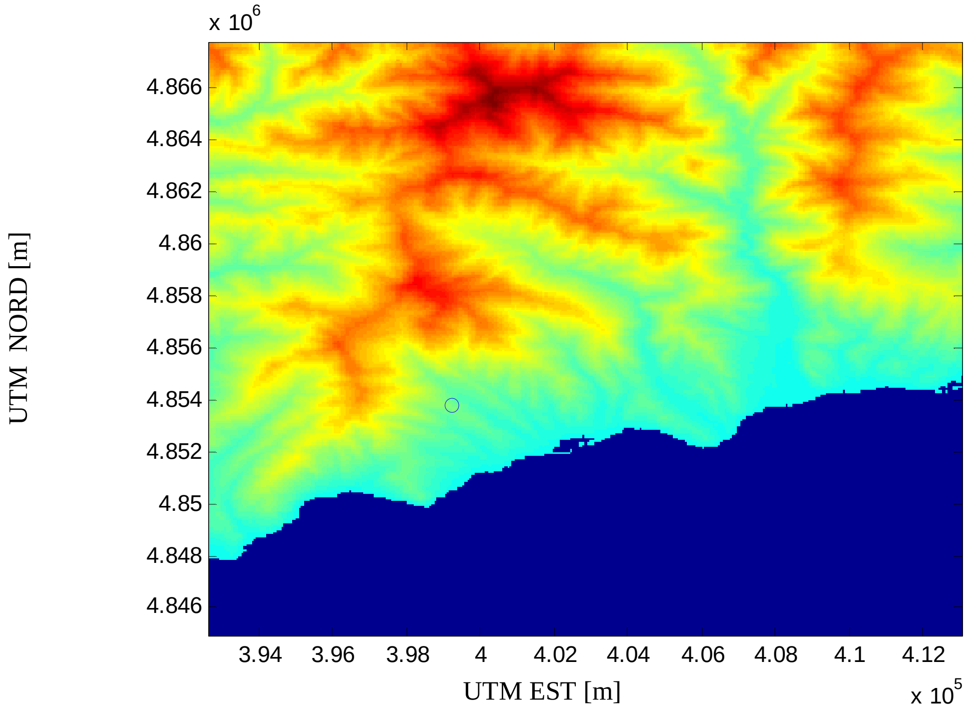


Figure 43c: DEM resized and georefered from strm after *leggiDEM_strm.m*

The resized and georefered DEM is a input file for *CostruzioneFOVaggiornato.m* and the parameters setting as North Orientation, inclination, horizon elevation, sensors position and number of scene are the input data for the Matlab file as well. The location of the sensor, in term of UTM coordinates, is: North UTM = 4853768 and EST UTM 399251 area 32T and the horizon elevation is 10 meters. Those parameters and data are necessary in order to calculate the cone of view on DEM and to consider the pixel and cell inside of it only.

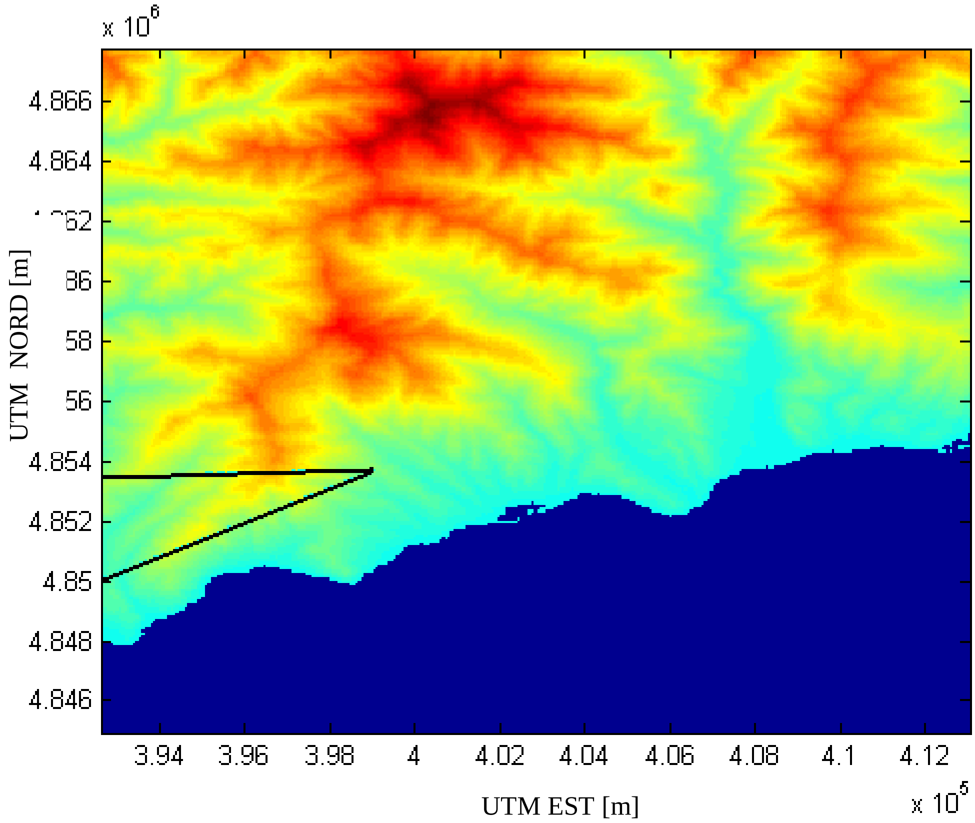
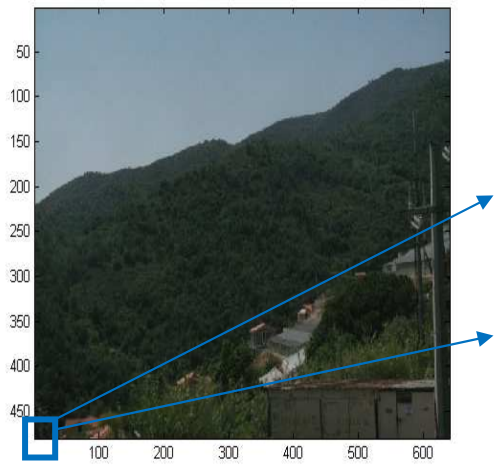


Figure 44: Field of view on DEM of Scene number 1.

In this function the horizontal field of view it is divided into portion equal to the y image resolution and the profile is extracted for each portion. At this point, the function *PixelToLatLong.m* provides to elaborate each profile subdividing it into portion equal to the horizontal image resolution as shown on figure 39. For the first image pixel considered is associated the first dotted line. When this line across the profile at least in one point, the Est UTM coordinate is linked with the image pixel. Otherwise, if the line does not across any point, the image pixel coordinate is linked as 0, because is related to a point in the sky.

At Est UTM point founded, for that portion of FOV and for that profile, is associated one point of North UTM. The operation is iterated a number of time equal to the image resolution, and every image pixel is linked as geographical coordinate. Finally we have two Matrix with NxM size where N are the rows and M are the columns of image. The function *ConvUtmToDeg.m* provides to convert the point found from UTM to Degree and the function WriteResultsFileTxt.m provides to generate the txt file where the position nxm corresponds to the pixel nxm on the image. In the figure below is shown an example.



474	7.751832	7.751859	7.751859	7.751859	7.751859	7.751832	7.751803
475	7.751803	7.751831	7.751831	7.751831	7.751831	7.751564	7.751489
476	7.751635	7.751803	7.751803	7.751803	7.751524	7.751444	7.751385
477	7.751385	7.751468	7.751468	7.751468	7.751385	7.751325	7.751280
478	7.751245	7.751301	7.751301	7.751301	7.751245	7.751206	7.751176
479	7.751106	7.751134	7.751134	7.751134	7.751106	7.751086	7.751071
480	7.750967	7.750967	7.750967	7.750967	7.750967	7.750967	7.750967

474	43.834159	43.834172	43.834172	43.834172	43.834172	43.834159	43.834146
475	43.834146	43.834159	43.834159	43.834159	43.834159	43.833908	43.833834
476	43.833980	43.834146	43.834146	43.834146	43.833869	43.833789	43.833730
477	43.833730	43.833813	43.833813	43.833813	43.833730	43.833670	43.833626
478	43.833591	43.833646	43.833646	43.833646	43.833591	43.833551	43.833522
479	43.833452	43.833480	43.833480	43.833480	43.833452	43.833432	43.833417
480	43.833313	43.833313	43.833313	43.833313	43.833313	43.833313	43.833313

Figure 45: example of a geo-ref. matrix latitude and longitude degree

In order to validate the algorithm some points are taken from the real monitored scenario using a GPS receiver. The real coordinates taken belong to some features which are well-seeing on image, for example the house, the three, poles etc. *CalibrationPoint.m* provides the point taken to the *DecisionSupportMethods.m* which calculate the differences and the statistics between the real point and the georefered point calculated trough the algorithm. This operation will be discuss more in details in the chapter related to the “*Results*”.

Finally the txt previous generated will be integrated in a graphical user interface (GUI) which include the smoke detection algorithm as well. In this way, automatically if an alarm is generated from smoke detection, the pixel alarmed will be georefered and additional support layer will be added to DEM or Google static Map, allows the early operation for fire extinguish. Even this will be discuss more in details in the chapter 5.