

AVNTK^{SC}



“RAPHAEL” GROUND CONTROL AND POST-PROCESSING STATION - GENERAL DESCRIPTION

**PRODUCED BY:
THE AEROSPACE TECHNOLOGY GROUP,
ARDITA AERONAUTICA DIVISION**

Technical POC:
Fidel Gutiérrez Resendiz, Ph.D.
Aerospace Remote sensing Group Leader
Guadalajara, Jalisco, Mexico
E-mail: fidel@ardita-aeronautica.com

Contents

1.	Summary	4
2.	Introduction	5
3.	Basic System Description	6
3.I	Introduction	6
3.II	Dynamic Mission Planning and Tasking	7
3.III	Synthetic Vision	8
3.IV	Interface with onboard avionics	9
3.V	Simulator & Autopilot	12
	Automatic Takeoff and Landing	12
	Takeoff Performance	15
	Stellar navigation	20
	Autopilot (helicopters/blimps/fixed wings)	21
	Control Panel	22
	Helicopter Dynamic Model	22
	Main rotor	22
	Blade calculation	23
	Bell-Hiller stabilizing bar	23
	Gravitation Model	24
	Physical Parameters	25
	Wind model	27
	Servo control input	27
	Atmosphere Earth Model	28
3.VI	Photogrammetry Module	28
	Camera Calibration	29
	Orthorectification	30
3.VII	Image restoration	32
	Increase Depth of Field	34
3.VIII	Raphael Geographical Information System	35
3.IX	Advanced GIS Analysis Modules	37
	Simulation - Cellular Automata -	37
	Simulation - Fire Risk Analysis -	40
	Simulation - Fire Spreading Analysis -	41
	Simulation - Hydrology -	42
	Simulation – Identification of unit hydrographs and component flows from rainfall, evaporation and streamflow data (IHACRES) –	44
	Simulation - Modelling the Human Impact on Nature -	46
	Terrain Analysis – Channels -	47
	Terrain Analysis – Hydrology -	49
	Terrain Analysis - Lighting, Visibility -	51
	Terrain Analysis – Morphometry -	53
	Terrain Analysis – Preprocessing	54
	Terrain Analysis – Profiles	55
	3D Visualisation	56

4.	Optional Modules	57
4.I	3D Reconstruction from Full Moving Video (FMV)	57
4.I.2.-	Camera Model and calibration	58
4.I.3.-	Detection Features	61
4.I.4.-	SIFT matching between pair images	62
4.I.5.-	Bundle Adjustment	64
4.I.6.-	3D Scene Surface Reconstruction	65
4.I.7.-	Poisson Surface Reconstruction	66
4.I.8.-	Textured 3D reconstructed scene	66
4.I.9.-	Alignment between independent mission tracks	68
	Comparison between CAD model and 3D reconstructed scene	69
	CAD Data vs. Multiple Images	70
4.II	Multispectral Analysis	71
	Multivariate Methods.	74
4.III	Satellite Data Access	79
5.	Standard “Raphael Ground Control Station”	81
6.	Pricing details	82
7.	Collaboration Structure	83
8.	Appendices	84
8.I	Detail of GIS Functions:	84
	Elementary grid operations.	84
	Interpolation of grids.	85
	Geostatistics	85
	Discretization	86
	Gridding	87
	Advanced grid operations I	87
	Advanced grid operations II	88
	Import and Export between raster/vector data formats.	89
	Geospatial Data Abstraction Library	90
	Import/Export Grids	92
	Image Import and Export	93
	Import and Export of vector data	93
	Tools for the import and export of tables.	93
	Shapes – Tools related to gridded and vector data	94
	Shapes – Tools for lines.	94
	Shapes – Tools for the manipulation of point vector data.	94
	Shapes – Tools for Polygons	95
	Shapes – Tools for the manipulation of vector data.	95
	Table – This module is designed for table calculations.	95
	Table – Tools	96
	TIN – Tools for triangular Irregular Network (TIN) processing.	96
	Point Cloud – Tools for points clouds.	96
	Point Cloud – Point Cloud Viewer	96
	Projection – Tools for coordinate transformation based on PROJ.4	96
	Projection – Tools for the georeferencing of spatial data (grids/shapes)	97
9.	General Terms and Conditions	98

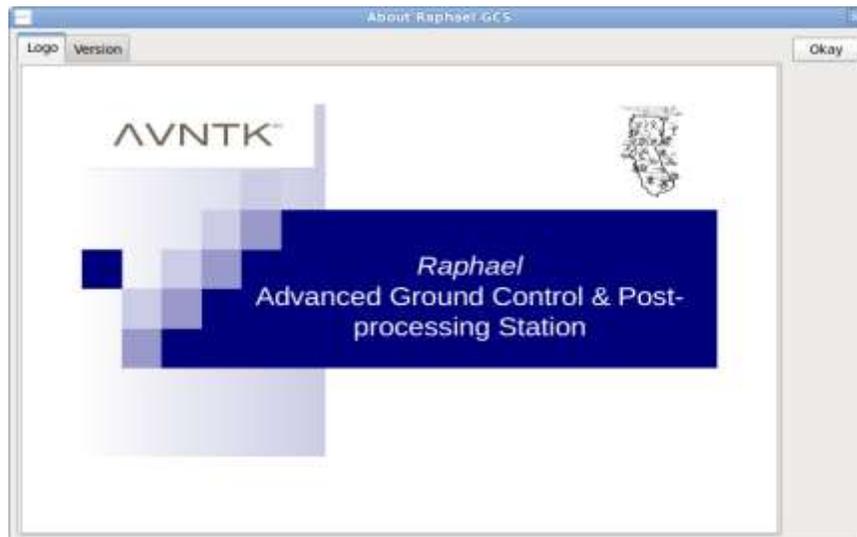
1. Summary

Raphael Ground Control Station (Raphael GCS) is an advanced system developed under LINUX operating system, it is based on C++ programming. **Raphael GCS** integrate several packages in only one system. For example, normal functions associated with a Ground Control Station are already implemented, such as mission planning /autopilot / remote control. **Raphael GCS** has integrate a scale helicopter simulator characterize by 80 physical parameters. It solves aerodynamic equations of a rigid body embedded on the earth gravitational potential. Synthetic vision module is a sophisticated virtual reality tool that is also implemented. The 3D synthetic vision is widely used on metadata visualization, for example multi-frequency satellite raster maps. Photogrammetry capability is also implemented on **Raphael GCS**. In addition, it is integrate an advanced metadata edition tool compatible with most popular geographic information systems software. Also, it can support the most widely used file formats for import/export metadata types.

Raphael GCS is powerful tool in pre-processing and post-processing stages, from basic image tasks until the most specialise functionalities, for example sum or differences between two grid sets until flight image-refining stage capable of improving UAV images sufficiently.

Our system has the most advanced modules such as autopilot, terrain analysis (channels, hydrology, lighting-visibility, morphometry, pre-processing, profiles), simulation (fire spreading analysis, fire risk analysis, hydrology analysis, human impact on nature), computer vision algorithm, multispectral data toolkit, spectral analysis (PCA, FastICA and multidimensional display), satellite access, 3D reconstruction from multiple images, coordinate transformations, metadata editing and 3D realistic vision are some of the capabilities of **Raphael GCS**.

We offer a product with the capability to produce the sort of information that civilian end-users, such as “real world” environmental/forestry/planning/building agencies require.



2. Introduction

Raphael GCS is the result of the realization that most UAV manufacturers have concentrated on hardware and functionality and not enough on post-processing of the data obtained. On the other hand, some potential users have advanced geographical information systems such as both ArcView or ArcGIS on windows and GRASS in a UNIX operating system, but these assume high input-image quality that is not easily obtainable from vibrating, low-resolution uncorrected UAV cameras, and cannot analyze UAV imaging, even with quite advanced GIS. Moreover, civilian customers require advanced analysis tools to derive useful results such as fire predictions, hydrology maps, cartography, 3D reconstruction from full moving video (FMV), build maps from interpolate random samples points, super resolution images, segmentations, transformation between multiple earth coordinate systems, visualization of 3D satellite metadata, multifrequency analysis, etc. Therefore, there is a need for a software tool capable of performing these tasks, to be combined with existing software, so as to offer an attractive product to potential civilian end-users.

We have developed a leading-edge ground control system for UAVs that includes a mission planning and re-tasking module, synthetic vision in a 3D virtual reality environment, an autopilot autonomous take-off and landing using a single camera, a robust wind-compensating algorithm, camera correction and calibration, a photogrammetry module, a complete geographical information system including modules such as orthorectification, georeferencing, image fusion, image mosaicing, ISO19,115-compliant metadata editing, satellite access, 3D visualization of multiple layers, multifrequency analysis, terrain analysis (channels, hydrology, lighting-visibility, morphometry, preprocessing, profiles), simulation (fire spreading analysis, fire risk analysis, hydrology analysis, Human impact on nature), computer vision algorithm, multispectral data toolkit, spectral analysis (PCA, FastICA and multidimensional display) and more.

This software was first shown at Farnborough 2008 and has been recently acquired by Mexican companies and a British company (The Goodwill Company Ltd.) for defense applications, and is being offered to UAV manufacturers worldwide. This station has been successfully employed to control fixed-wing, blimps and rotary-wing unmanned vehicles in applications such as power-line monitoring, surveillance and marine SAR. Current work includes enhancements such as the ability to analyze multi-spectral images, 3D reconstruction derived from real-time video, and persistent wide area surveillance.

Our system has been developed from modules originally produced for industrial applications, where throughput is high, the operator often undergoes fatigue, repetitive tasks and typically has a relatively low degree of specialization so that interfaces need to be user-friendly. This is somewhat similar to soldier interfaces of advanced technology that must be simple, using hints and pointing methods for the user. Clearly, good spatial abilities are also important in reporting because of the need to translate activity from the viewpoint of the sensor to that of personnel on the ground, whereas we had developed pretty much the same sort of capabilities but for manufacturing processes.

In recent years we have extended this capability to customised sensor and imaging exploitation for monitoring/surveillance purposes. At the moment, we are also building a high-power pylon multi-spectral monitoring system deployed aboard a blimp for the Mexican electricity Board and UAV micro-avionics compatible with *Raphael GCS* to be offered soon. The following is a brief description of the toolkit and its current capabilities.

3. Basic System Description

3.1 INTRODUCTION

The *Raphael GCS* GUI (Graphical User Interface) represents the linking element between the user and the *Raphael GCS* modules. The GUI has simple structure that allows working with many different data sources and results, while keeping all of them correctly organized.

The main *Raphael GCS* window, has a look like the one shown below these lines, see figure 3.1.1.

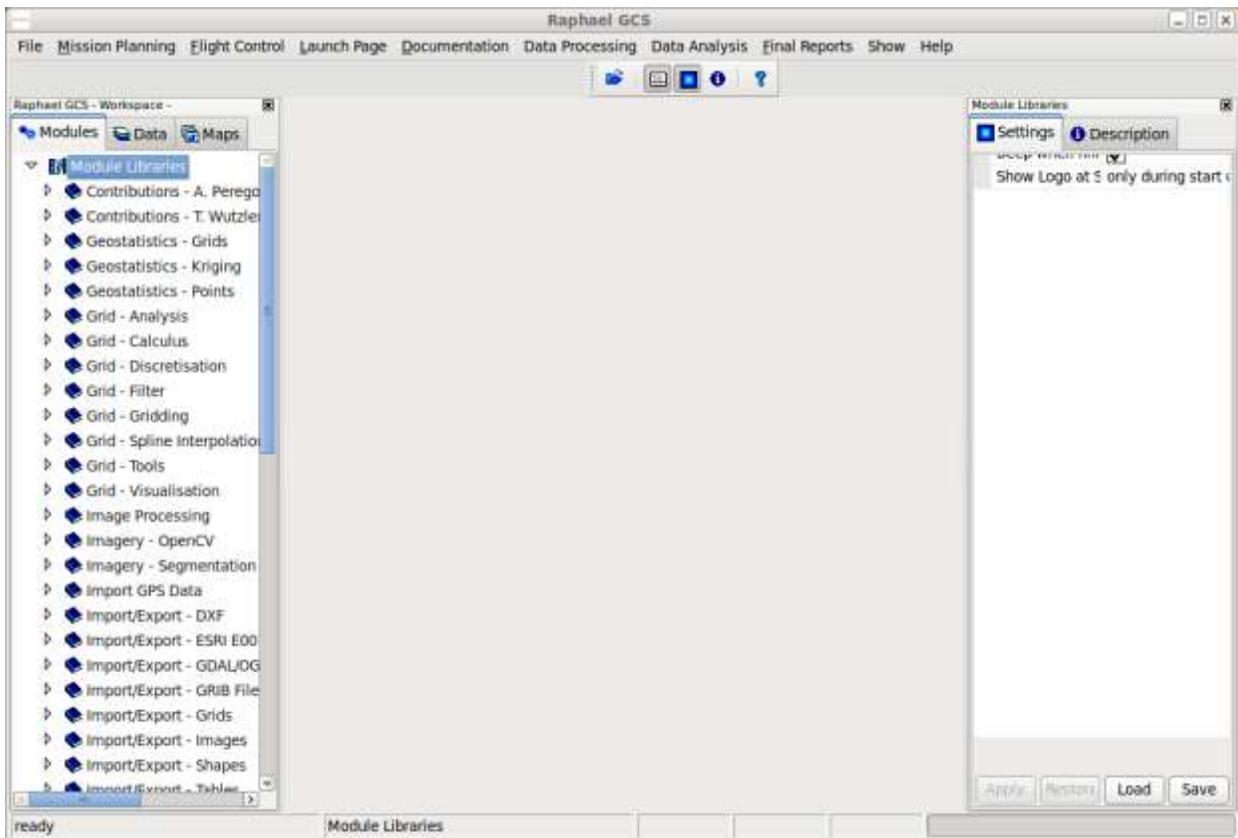


Figure 3.1.1: The *Raphael GCS* main window. It shows the graphical user interface.

The whole functionality of *Raphael GCS* can be found from this main window.

In the following sections, we present the various modules in more detail.

3.II DYNAMIC MISSION PLANNING AND TASKING

Today's high altitude endurance (HAE) reconnaissance unmanned aerial vehicles (UAVs) are extremely complex and capable systems. They are only as good as the quality of their implementation, however. Mission planning is rapidly increasing in complexity to accommodate the requirements of increasing aircraft and information control capabilities. Effective mission planning is the key to effective use of airborne reconnaissance assets, which demand extremely intensive and detailed mission planning.

The mission plan must accommodate a range of possible emergencies and other unplanned in-flight events, like pop-up threats or a critical aircraft system failure. Current in-flight mission re-planning systems do not have sufficient capability for operators to effectively handle the full range of surprises commonly encountered in flight operations. Automation is commonly employed to reduce this high workload on human operators.

Our dynamic mission planning module overcomes a variety of common operational situations in HAE UAV reconnaissance that necessitate more direct human involvement in the aircraft control process than is currently acknowledged or allowed. A state of the art mission planning software package, OPUS, can be used to demonstrate the current capability of conventional mission planning systems. This current capability can be extrapolated to depict the near future capability of highly automated HAE reconnaissance UAV in-flight mission replanning. Many scenarios exist in which current capabilities of in-flight replanning fall short.

Our dynamic mission planning module has been developed and implemented in *Raphael GCS* and when the same problematic scenarios are revisited with it, improved replanning results can be demonstrated, particularly being able to reroute in the light of new information and threats, slack time available, interpretation rating scale of points of interest and a given route survivability estimate. Capabilities include:

- Survivability estimate
- Threats
- Sensor dependence
- Imaging quality
- Route details
- Re-planning limitations

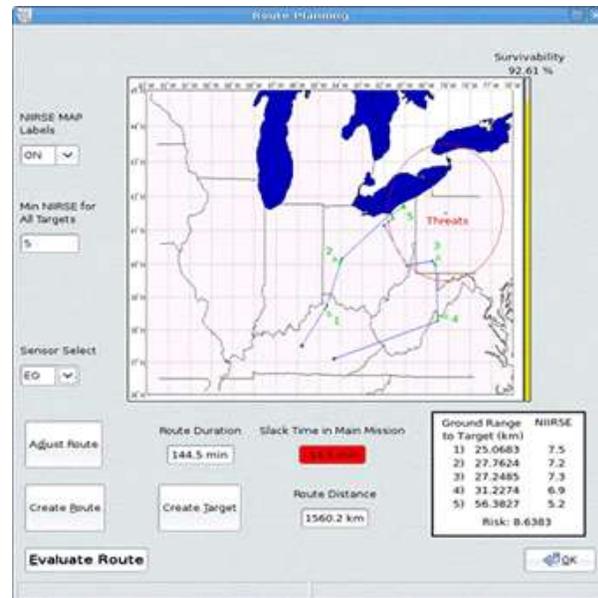


Figure 3.II.1: Route Planning GUI.

3.III

SYNTHETIC VISION

Raphael GCS has an embedded 3D virtual reality environment. This module can capture the general context of each position on earth surface. The user can choose any location on earth and it will show its topology under user-friendly environment. The 3D environment includes stellar information, solar positioning so shading and illumination conditions can be predicted and accounted for, as well as an extensive library of 3D objects. In summary, the 3D virtual reality environment has the following items:

- Advanced world topological DEM.
- Buildings
- Roads
- Rivers
- lakes
- Vegetation (trees, grass)
- Urban furniture
- cars
- transit signals
- Textures
- Snow
- Sky
- Water
- Fog

The synthetic vision available on *Raphael GCS* has typical screenshot given by the figure 3.III.1.



Figure 3.III.1: Typical screenshots using the synthetic vision available on *Raphael GCS*, with and without navigational aids.

Raphael GCS has implemented numerous of computer vision algorithms, such as the Fourier Transform, Feature Detection, Motion Analysis, Object Tracking, Canny detector, Harris detector, Hough transform for line detector, Hough transform for circle detector, SIFT algorithm, Calculate the optical flow for a sparse feature set using the iterative Lucas-Kanade method with pyramids, Compute dense optical flow using Gunnar Farneback's algorithm, a function implementing the CAMSHIFT object

tracking algorithm, etc. All these algorithms are based on openCV as well as Vigna software, for example object tracking, for which we have implemented a tracking algorithm to follow an objective. Feature detection and match correspondence is a crucial step to make a 3D reconstruction model. By itself, tracking is an essential step in the mapping of objectives.

The virtual reality environment has a number of 3D libraries to help generate realistic environments including a range of surface textures (bricks, cement, doors, tiles, windows, wood, etc.), vegetation primitives including a wide range of trees, city items such as lights and signals, cars, posts including high-voltage posts, and the like.

In global terms, the environment correctly places the Sun in relation to the Earth at any one time and any geographical DEM position can be freely downloaded from <http://www.gdem.aster.ersdac.or.jp> with a resolution of 30 meters, and used as a reference for UAV-derived imaging, comparison with satellite data, or many other forms of GIS information processing.

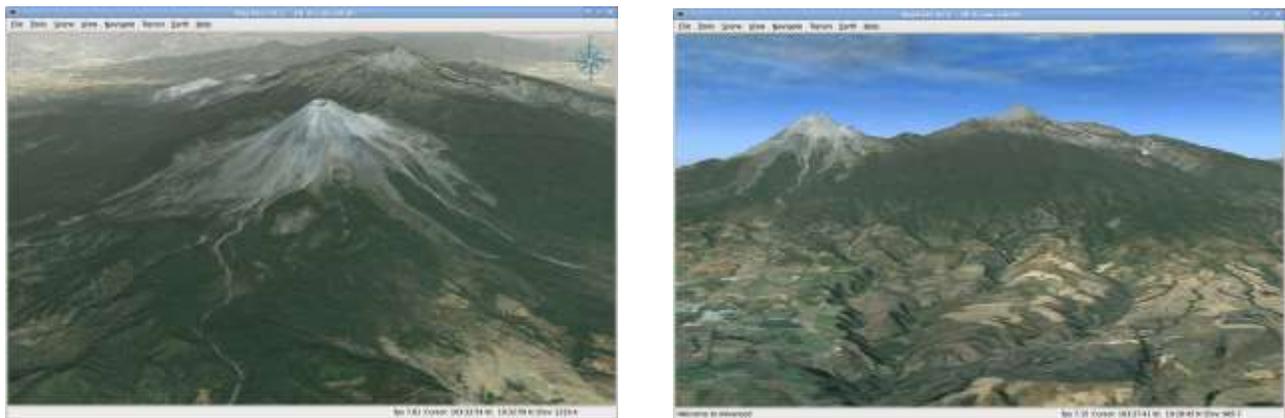


Figure 3.III.2: The Colima Volcano in Jalisco Mexico, using synthetic data, i.e. we have used a DEM with 30 meters resolution and Raphael GCS texture.

3.IV INTERFACE WITH ONBOARD AVIONICS

Our ground control station has been used with a wide range of unmanned vehicles including fixed-wing UAVs, blimps and helicopters.

A range of communication protocols are available and have been tested including WiFi (using Matchport hardware), GSM-GPRS (using Fargo Maestro-100 hardware), and RF. High Speed Download Packet Access (HSDPA) is also an option but not yet tested because it is too expensive to implement due to the costs of the hardware and the monthly subscription costs for the use of 3G services. As a commercial large-scale implementation rather than small-scale vehicles, this would be the ideal choice of communications standard due to its high data rates and large range. The large coverage is due to the wide availability of 3G, it is available in most urban areas making it ideal for surveillance use.

RF is cheap to implement and has a range of up to 10 km. The range is large enough to control a quadcopter for instance as the battery life is only 10-15 minutes typically, which limits the distance that it can cover. The main problem with RF is that it is not as readily available as other standards, so only covers the distance from an RF transmitter. Another issue is that the data rate is only up to 10

kbps which is not high enough for good quality video transmission. The MikroKopter is equipped with a basic RF receiver that can be used for control, but was not be upgraded for data communications.

GSM (GPRS) is cheap to implement due to the maturity of the technology. The range of GPRS is large as it works over the cellular network so will work anywhere there is a GSM signal, most urban areas have very good signal quality. The data rate of the GPRS class 10 modems is 16-24kbps upload and 32-48kbps download. In theory this will be large enough to send a decent quality video stream. GPRS offers the ideal balance of range, cost and availability.

WiFi is a good choice as proof of concept, as it is relatively cheap to implement and has a wide enough range for testing purposes. The data rate for the WiFi transceiver is 11Mbps which is large enough for good quality video transmission. For proof of concept, this should be used before GPRS is implemented as it has a wide enough range for testing. WiFi is more suited to close range applications and is therefore easier to demonstrate at a Trade Fair for example.

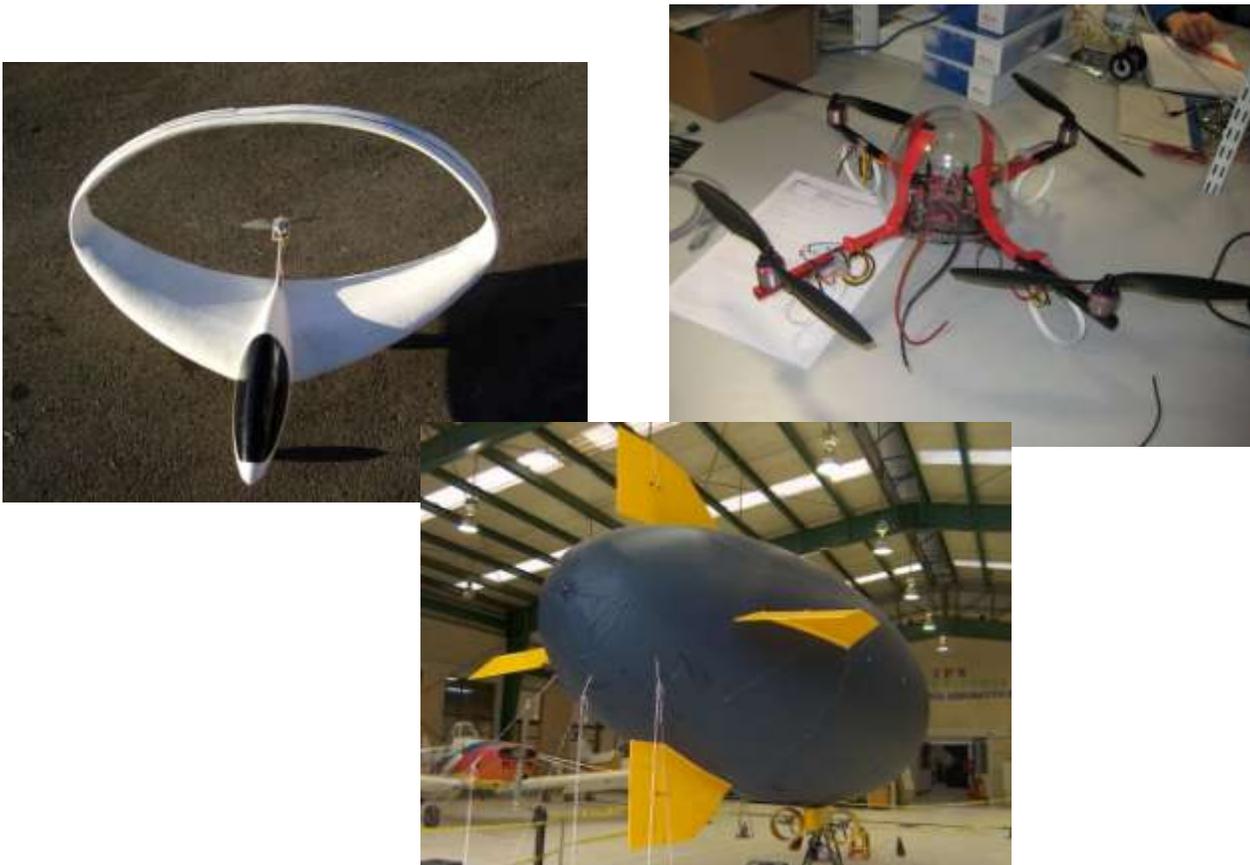


Figure 3.IV.1. Some of the platforms that *Raphael GCS* has been configured to control.

The control panel includes all the typical functions found in standard GCS platforms. It is split into two screens, a flight-control screen and a visual onboard camera-driven screen.



Figure 3.IV.2. *Raphael* GCS UAV control panel and onboard camera view (with all optional displays on).

Automatic Takeoff and Landing

Raphael GCS has implemented a strategy for an autonomous both takeoff and landing of an Unmanned Aerial Vehicle for both rotor wings and fixed wings configuration.

Given that autonomous vehicles such as underwater vehicles, aircrafts, and helicopters are highly non-linear dynamic systems. The challenges involved in the design of a control strategy for such dynamic system, there exists the problem of accurate position measurement in such machines. Flying machines are usually equipped with on-board inertial sensors which only measure the rate of motion. The position is thus obtained from time integration of rate data, resulting in potential drift over time due to sensor noise. To overcome this problem, *Raphael GCS* uses vision sensors and computer vision algorithms within the feedback loop control system. The strategy for the autonomous takeoff/landing maneuver is using data taken from the position information system obtained from a single monocular on-board camera and inertial measurement unit (IMU). Thus, the proposed vision-based control algorithm is built upon homography-based techniques and Lyapunov design methods in order to estimate position, velocity and attitude of the flying machine during its navigation.

Without loss of generality the following method can be seen as takeoff phase or landing phase. We take landing point of view to describe the methodology of autonomous control.

Homography determination of position and pose during landing approx.

The UAV is assumed to be equipped with Inertial Measurements Unit (IMU) from which velocity information can be deduced. A homography-based approach has been utilized for the determination of position and pose of the UAV with respect to the landing pad.

The homography-based approach is well suited general application since all visual features are embedded on a flat planar landing pad. On the other hand, a constant design vector is integrated within the filtered regulation error signal, resulting in an input matrix that facilitates an advantageous coupling of translation dynamics of the UAV to the rotational torque inputs. Additionally, the null space of this input matrix is used to achieve a secondary control objective of damping the orientation error signal of the UAV to within a neighborhood about zero which can be made arbitrarily small through the proper selection of design parameters.

In the next section we will present a brief discussion of the camera projection model and then introduce the homography relations, further camera model and camera calibration see section 3.VI and section 4.I.

Projection models

Visual information is a projection from the 3D world to the 2D camera image surface. The pose of the camera determines a rigid body transformation from the current camera fixed frame B to the reference frame I and subsequently from the desired image frame B_d to I . One has

$$\mathcal{X}_I = R \mathcal{X} + \xi \quad (1)$$

$$\mathcal{X}_I = R_d \mathcal{X}_d + \xi_d \quad (2)$$

as a relation between the coordinates of the same point in the current body fixed frame ($\mathcal{X} \in B$) and the desired body frame ($\mathcal{X}_d \in B_d$) with respect to the world frame ($\mathcal{X}_I \in B_I$). And where ξ and ξ_d are expressed in the reference frame I .

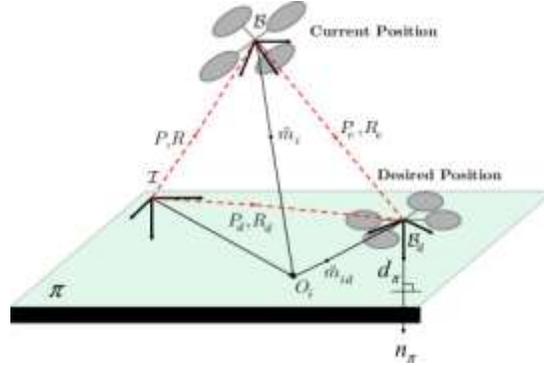


Figure 3.V.1: It shows relationship between inertial and body fixed coordinates frames for a UAV on a landing approach.

Note: There are 2 kinds of projection used in vision: the spherical and the flat projection. The spherical projection identifies the projection plane as the spherical surface and the image point p is given by $p = \frac{1}{\chi} [xyz]$. However, in the flat projection the point is projected on a plane with its image $p = \frac{1}{z} [xyz]$. Indeed, since equality in projective geometry is an equality between directions, both points are on the same ray emanating from the origin and are thus not distinguished. **Raphael GCS** will assume a calibrated camera but we do not distinguish between spherical or flat projections.

Planar Homography

Let $\bar{m}_i \in \mathbb{R}^2$, $\bar{m}_{id} \in \mathbb{R}^2$ denote the Euclidean coordinates for the i^{th} visual feature O_i on the landing surface relative to the camera at position B and B_d , respectively. From the geometry between the coordinate frames, $\bar{m}_i \in \mathbb{R}^2$, $\bar{m}_{id} \in \mathbb{R}^2$ are related as follows

$$\bar{m}_i(t) = P_e + R_e \bar{m}_{id} \quad (3)$$

Also illustrated in figure 3.V.1, $n_\pi \in \mathbb{R}^3$ denotes the known constant normal to the plane π expressed in the coordinates of B_d , and the constant $d_\pi \neq 0 \in \mathbb{R}^1$ denotes the distance of the landing surface

π from the origin of the frame B_d . It can be seen from figure X that for all i visual features, the projection of \bar{m}_{id} along the unit normal n_π is given by

$$d_\pi = n_\pi^T \bar{m}_{id} \quad (4)$$

Using equation (4), the relationship in eq. (3) can be expressed in the following manner

$$\bar{m}_i = \underbrace{\left(R_e + \frac{1}{d_\pi} P_e n_\pi^T \right)}_H \bar{m}_{id} \quad (5)$$

where $H \in \mathbb{R}^{3 \times 3}$ represents a Euclidean Homography. To express the above relationship in terms of the measurable image space coordinates of the visual features relative to the camera frame, the normalized Euclidean coordinates m_i , $m_{id} \in \mathbb{R}^3$ for the visual features are defined as

$$m_i = \frac{\bar{m}_i}{z_i}, \quad m_{id} = \frac{\bar{m}_{id}}{z_{id}} \quad (6)$$

where z_i and z_{id} are the third coordinate elements in the vector \bar{m}_i and \bar{m}_{id} , respectively. The 2D homogeneous image coordinates of the visual features, denoted by p_i , $p_{id} \in \mathbb{R}^3$, expressed relative to B and B_d , respectively, are related to the normalized Euclidean coordinates by the pin-hole model camera such that

$$p_i = A m_i, \quad p_{id} = A m_{id} \quad (7)$$

where $A \in \mathbb{R}^3$ is a known, constant, upper triangular and invertible intrinsic camera calibration matrix. Hence the relationship in (5) can now be expressed in terms of image coordinates of the corresponding feature points in B and B_d as follows

$$p_i = \underbrace{\frac{z_{id}}{z_i}}_{\alpha_i} \underbrace{A \left(R_e + \frac{1}{d_\pi} P_e n_\pi^T \right) A^{-1}}_G p_{id} \quad (8)$$

where $\alpha_i \in \mathbb{R}^1$ denotes the depth ratio. The matrix $G \in \mathbb{R}^{3 \times 3}$ in (8) is a full rank homogeneous colineation matrix defined up to a scale factor, and contains the motion parameters P_e and R_e between the frames B and B_d .

Given pairs of image correspondences $[p_i, p_{id}]$ for four feature points O_i , at least three of which are non-collinear, the set of linear equation in (8) can be solved to compute a unique G up to a scale factor. When more than four feature point correspondences are available, G can also be

recovered (again, up to a scale factor) using techniques such as least-squares minimization. G matrix can then be used to uniquely determine H , taking into account its known structure to eliminate the scale factor, and the fact that the intrinsic camera calibration matrix A is assumed to be known. By utilizing epipolar geometry among many others methods, H can be decomposed to recover the rotational component R_e and the scaled translational component $\frac{1}{d_{pi}} P_e$.

In summary *Raphael GCS* has two modules about automatic Takeoff and Landing of UAV.

Takeoff: This module takes the craft from the current position to the First Waypoint Before Autopilot (FWBA) position from which the autopilot takes over. This point needs to be specified in the “Mission Planning” module.

Landing: This module receives craft control from the autopilot and takes the craft from the Last Waypoint Before Landing (LWBL) position to an end of runway position from which the operator takes over. This point needs to be specified in the “Mission Planning” module.

Takeoff Performance

Many standards are used to define the stages of an aircraft takeoff run, depending on the country and type of aircraft. We have taken the Federal Aviation Regulation (FAR) for illustration propose to use the definition of takeoff. Under FAR 25, an aircraft taking off performs a ground roll to rotation velocity, rotates to liftoff attitude, lifts off and climbs to a height of 35 ft. This definition can be applied to two types of takeoff: takeoff with all engines operating (AEO) and takeoff with engine failure, usually prescribed as one engine inoperative (OEI). Each of these types of takeoff will be discussed in turn.

Takeoff with all engines operating is the type dealt with in most day-to-day situations. The aircraft accelerates from a stop or taxi speed to the velocity of rotation, V_r , rotates to the liftoff attitude with corresponding velocity V_{lo} , and climbs over an obstacle of 35 feet as shown in figure X. The velocity at the end of the 35 ft climb is usually called the takeoff safety speed and given the designation V_2 . FAR 25 prescribes limits to these velocities based on the stall velocity, V_s , the minimum control velocity, V_{MC} , and the minimum unstick velocity, V_{MU} . These three velocities are physical minimum velocities under which the aircraft can operate.

The stall velocity is the aerodynamically limited velocity at which the aircraft can produce enough lift to balance the aircraft weight. This velocity occurs at the maximum aircraft lift coefficient, C_{Lmax} , and is defined as:

$$V_s = \sqrt{\frac{2(W/S)}{\rho C_{Lmax}}}$$

The minimum control velocity V_{MC} is the “lowest airspeed at which it has proved to be possible to recover control of the airplane after engine failure”. This is an aerodynamic limit which is difficult to predict during the preliminary design stage, but may be obtained from wind tunnel data during later phases of design. The minimum unstick velocity V_{MU} is the “airspeed at and above which it can be demonstrated by means of flight tests that the aircraft can safely leave the ground and continue takeoff”. This velocity is usually very close to the stall velocity of the aircraft.

With these reference velocities defined, the FAR places the following broad requirements on the velocities of takeoff:

$$\begin{aligned} V_r &\geq 1.05 V_{MC} \\ V_{lo} &\geq 1.1 V_{MU} \\ V_2 &\geq 1.2 V_{MC} \end{aligned}$$

Because the minimum unstick velocity is usually very close to the stall velocity the liftoff velocity is often referenced as greater than 1.1 times the stall velocity, rather than 1.1 times the minimum unstick velocity.

Engine failure brings another level of complexity to the definitions and requirements of takeoff. Usually takeoff of this nature is categorized by the failure of one engine, or one engine inoperative (OEI). OEI takeoff includes an AEO ground run to engine failure, an OEI ground run to liftoff, and a climb to 35 ft, also with OEI as illustrated in figure X. A takeoff with an engine out will take a longer distance than AEO takeoff due to the lower acceleration produced by the remaining engines. The obvious questions to ask are if the OEI takeoff field length required is longer than the field length available and if the distance to brake to a stop after engine failure is longer than the available field length. These questions are often answered by solving for a critical or balanced field length (CFL or BFL); the distance at which OEI takeoff distance equals the distance needed to brake to a full stop after engine failure.

Defining the CFL leads back to the time, or more specifically, the velocity at which engine failure occurs. As it turns out, by imposing the CFL definition, there is an engine failure velocity which uniquely defines the critical field length. This velocity is often called the critical velocity, V_{crit} . It must be noted that during an aborted takeoff some amount of time will be required after the engine fails for the “pilot” to actually begin braking, both because of the pilot’s reaction time and the mechanics of the aircraft. During this passage of time, the aircraft continues to accelerate on the remaining engines and will finally reach the decision velocity, V_1 .

Careful inspection of the above definitions will show that engine failure at a velocity lower than the critical velocity will require an aborted takeoff, while engine failure after the critical velocity has been reached will require a continued takeoff. With the above definitions in place, FAR 25 imposes additional requirements for OEI takeoff:

$$V_r \geq V_l$$

$$V_{lo} \geq 1.05 V_{MU}$$

Note that although other standards for aircraft takeoff exist, most use the same four velocities in their takeoff analysis: V_1, V_r, V_{lo}, V_2 .

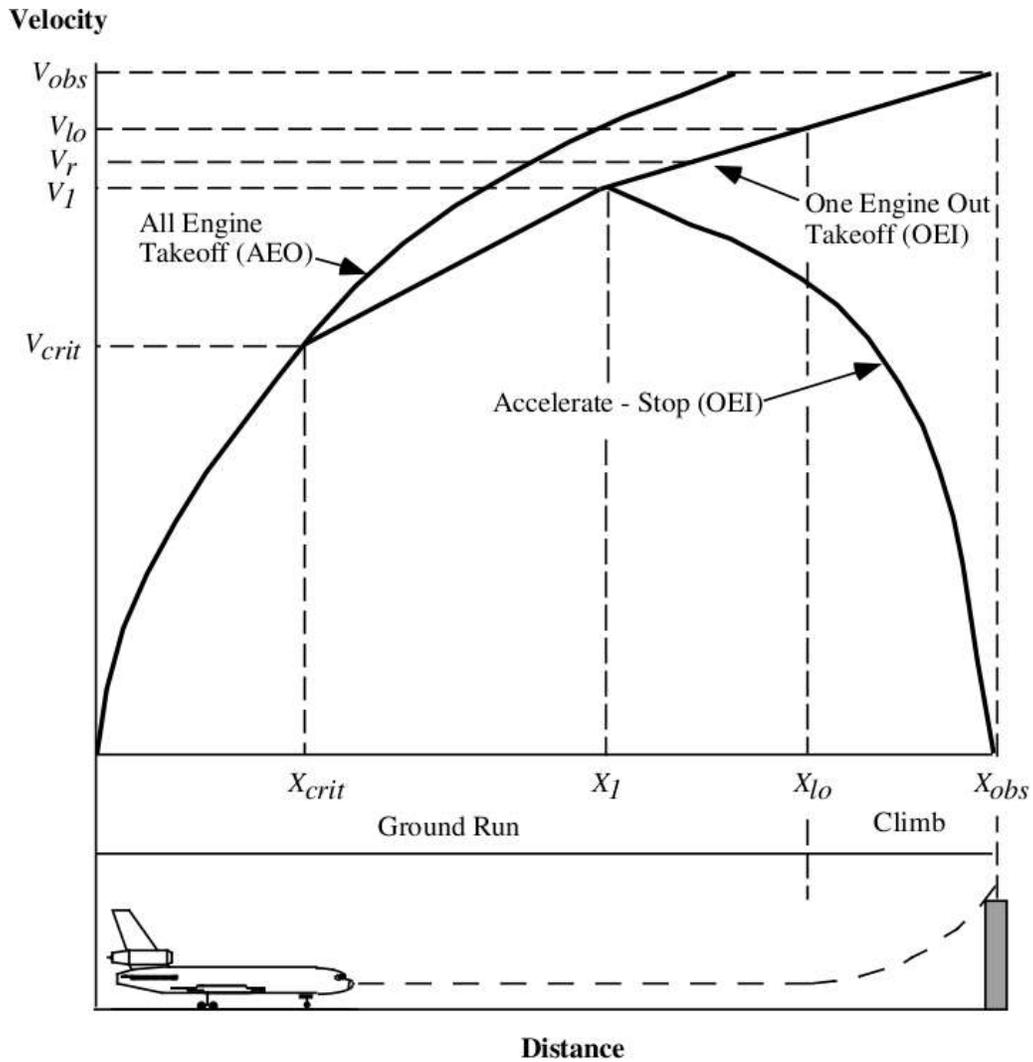


Figure 3.V.2: A normal takeoff involves a definition of each velocity and distances.

Raphael GCS has implemented the simplified method proposed by Powers, and a modified version of a method proposed by Krenkel and Salzman. The simplified Powers method requires 13 input parameters and solves the governing equation analytically for takeoff times and distances. The method assumes constant thrust throughout the takeoff run and climb phase aerodynamics are the same as in the ground roll. The two major problems with the Powers methodology are the lack of a rotation phase and the use of ground roll equations of motion to predict the climb phase of takeoff. The lack of a rotation phase causes the method to under predict ground roll at times and the climb phase is often over predicted. The modified Krenkel and Salzman method requires 25 input parameters and allows thrust vectoring and assumes thrust varying with velocity. A modification was made to assume thrust varied quadratically with velocity. Originally, the method solved the equations of motion, both nonlinear ordinary differential equations, parametrically. Due to consistent under prediction of the ground roll, the method was also modified to include a rotation phase; a continuation of the ground roll for a user-defined amount of time. As with the simplified Powers method, the modified Krenkel and Salzman method iterated from an initial guess critical engine failure velocity to predict the BFL. Unlike the Powers' method, the Krenkel and Salzman method increased the engine out rotation velocity to allow the aircraft to take off with reduced thrust.

Takeoff Example

Name	Value	Unit
Atmospheric density	0	sl/ft ²
Aircraft Weight	95000	lbs
Wing area	1000	ft ²
Maximum lift coefficient	2	
Ground lift coefficient	0.3	
Air lift coefficient	1.65	
Ground drag coefficient	0.08	
Air drag coefficient	0.121	
Rolling friction coeff.	0.025	
Braking friction coeff.	0.3	
Angle of thrust	0.00	rad
Stall margin	1.1	
Descision time	3.00	sec
Obstacle height	35.00	ft
OEI power remaining	0.5	

$$\text{Thrust} = 31450.0 - 17.263404 * V + 0.025019 * V^2$$

Normal Takeoff:

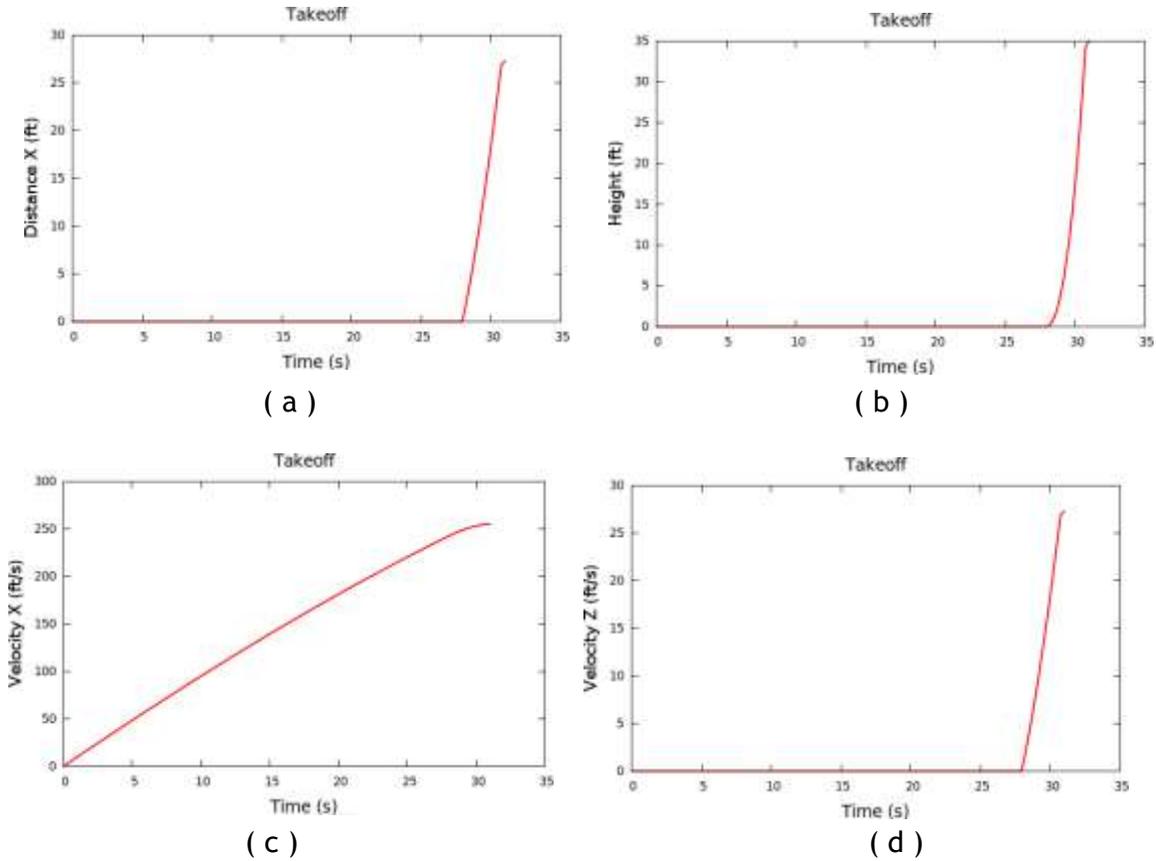


Figure 3.V.3: It shows the normal takeoff of a DC9 aircraft.

Normal Take-off Summary

Name	Value	Unit
Rotation Velocity	219.912	ft/s
Lift-off velocity	242.080	ft/s
Velocity over obstacle	256.451	ft/s
Rotation distance	2862.147	ft
Lift-off distance	3555.391	ft
Distance to obstacle	4260.314	ft
Rotation time	24.936	s
Lift-off time	27.936	s
Time to obstacle	30.759	s

OEI Take-Off Summary

Name	Value	Unit
Critical Velocity	203.886	ft/s
Decision Velocity	212.381	ft/s
Velocity over obstacle	233.529	ft/s
Critical Distance	2419.709	ft
Decision Distance	3044.195	ft
Balanced Field Length	5398.148	ft
Critical Time	22.849	s
Decision Time	25.849	s
OEI Takeoff Time	36.300	s

Stellar navigation

The on-board camera can be used as a star sensor for star image acquisition system. The algorithm employed is to determine star identification. Raphael GCS has several algorithms for star identification, a) correlation of known star position from an empirical star catalog with the unidentified stars in the star image. b) Pyramid algorithm, c) triangulation algorithm. Once the stars in an image have been identified orientation and attitude can be inferred based on which stars are in view and how these stars are arranged in the star image.



Figure 3.V.4: It shows the interface for stellar navigation, available on *Raphael GCS*.

Star location data is available from the 5th revised edition of the Yale Bright Star Catalog (1991) in celestial coordinates, which are a kind of polar coordinates. Setting a maximum magnitude (minimum brightness) of: 5.5 gives 2887 stars, while 5.8 gives 4103 stars. As these magnitudes are barely visible under the best of conditions, one can see that a relatively small number of stars is sufficient to draw an accurate sky.

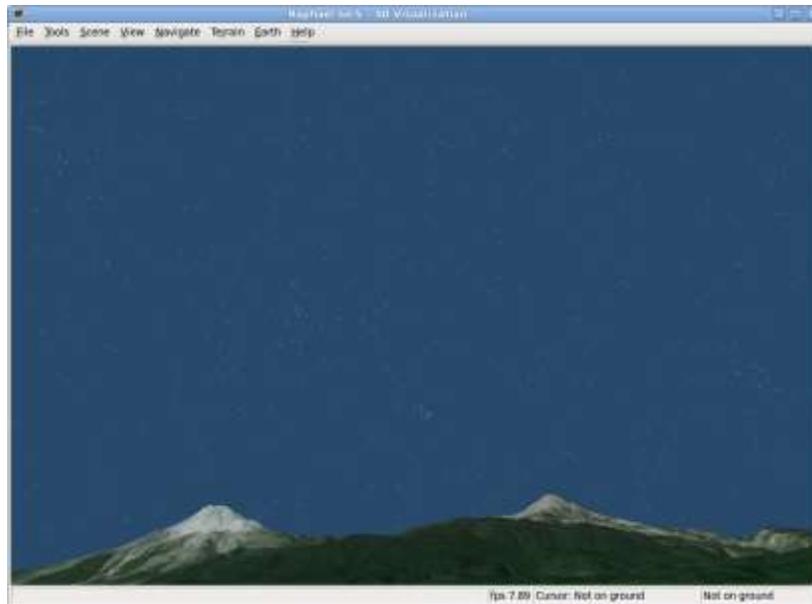


Figure 3.V.5: A sample night sky available on *Raphael GCS*.

Autopilot (helicopters/blimps/fixed wings)

The autopilot was tested on fixed-wing aircraft and blimps but undoubtedly the most demanding application is that of a helicopter. *Raphael GCS* has implemented a sophisticated 80-parameter helicopter simulator. We have modelled this non-linear dynamic system using advanced routines that calculate in detail every component of our own flight simulator. In the following sections, we describe the dynamics of the system using conventional six degree of freedom rigid body model driven by forces and moments that explicitly includes the effects of the main rotor, stabilizing bar, tail rotor, fuselage, horizontal tailplane, vertical fin and gravity. Also, it includes a module to make the algorithm robust and linearised. The control is effected through a Kalman filter as standard.

The model used for the atmospheric parameters such as temperature, pressure, density, speed of sound, gravity, etc. is that of the NASA 1976 standard atmosphere with a variation of less than 1% compared to tabulated values. The autopilot model includes models of the systems response of the two types of servo used in our UAV hardware.

The autopilot is able to navigate in three modes: GPS, inertial and stellar. For the stellar navigation, a basic set of bright stars is used, based on the 5th revised edition of the Yale Bright Star Catalog, 1991. The star tracking algorithm is a high accuracy proprietary method able to resolve position and time of a craft anywhere in the world down to 100 meters.

Control Panel

The control panel for the simulator is identical to that on the UAV control module, as the graphical interface between the flight simulator and the user. It displays useful information about simulation state like: Ground Speed, Altimeter, Trimming, Battery, Coordinates, Air Temperature, Compass.

Helicopter Dynamic Model

Figure 3.VI.6 shows the general structure of the helicopter model, where f_g is the gravitational force, f and n the remaining external force and moment vectors, respectively, and $u = [\delta_0 \delta_{1c} \delta_{1s} \delta_{0t}]$ the command vector that consists of the main rotor collective input δ_0 , main rotor and flybar cyclic inputs δ_{1c} and δ_{1s} , and tail rotor collective input δ_{0t} .

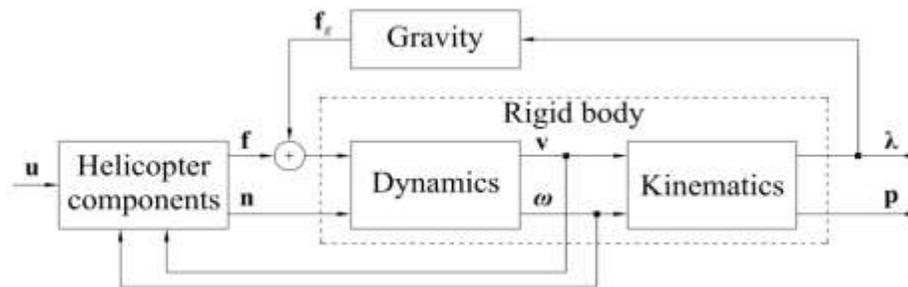


Figure 3.VI.6. Helicopter dynamic model - block diagram -

The total force and moment vectors account for the contributions of all helicopter components, and can be decomposed as

$$f = f_{mr} + f_{tr} + f_{fs} + f_{tp} + f_{fn}$$

$$n = n_{mr} + n_{tr} + n_{fs} + n_{tp} + n_{fn}$$

where subscript *mr* stands for main rotor, *tr* for tail rotor, *fs* for fuselage, *tp* for horizontal tailplane, and *fn* for vertical fin. As the primary source of lift, propulsion and control, the main rotor dominates helicopter dynamic behaviour. The Bell-Hiller stabilizing bar improves the stability characteristics of the helicopter. The tail rotor, located at the tail boom, provides the moment needed to counteract the torque generated by the aerodynamic drag forces at the rotor hub. The remaining components have less significant contributions and simpler models as well. In short, the fuselage produces drag forces and moments and the empennage components, horizontal tailplane and vertical fin, act as wings in forward flight, increasing flight efficiency.

Main rotor

A rotary-wing aircraft flies via an engineering process called blade element theory which involves breaking the main rotor and tail rotor in many small elements and then finding forces, trust, torque

and power. The main rotor is not only the dominant system, but also the most complex mechanism. It is the primary source of lift, which counteracts the body weight and sustains the helicopter on air. Additionally, the main rotor generates other forces and moments that enable the control of the aircraft position, orientation and velocity. *Raphael GCS* uses the rotor dynamic model, whose main building blocks are depicted in figure 3.VI.7.

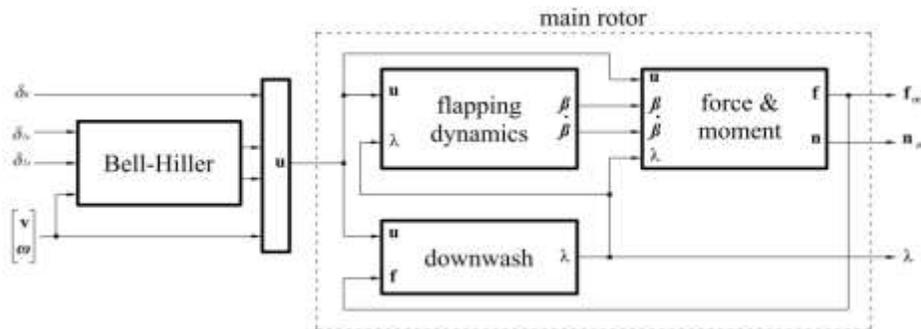


Figure 3.VI.7: Main rotor block diagram.

Blade	
a	Lift curve slope (1/rad)
R	Blade radius (ft)
ω	Rotor angular velocity (rad/s)
b	Number of rotor blades
c	Rotor blade chord (ft)
R_0	Blade root cutout (ft)
θ_c	Blade collective pitch at 75% R (rad)
θ_{twist}	Blade twist for washout (rad)
Cd_0	Profile drag coefficient
e	Oswald efficiency factor
V_{\perp}	Perpendicular velocity to rotor disk in direction of T (ft/s)
ρ	Air density (slug/ft ³)
T	Output thrust (lbs)
Q	Output torque (lb-ft)
P	Output power (lb-ft/s)
avg_{v_1}	Output average induced velocity (ft/s)

Table 3.VI.1: Structure for the blade element calculations.

Blade calculation

In order to calculate thrust, power, torque and induced velocity of the main rotor and tail rotor, we have used a combined blade element moment theory. The table 3.VI.1 shows the structure for these calculations.

Bell-Hiller stabilizing bar

Currently, almost all model-scale helicopters are equipped with a Bell-Hiller stabilizing bar, a mechanical blade pitch control system that improves helicopter stability. From a control point of view, the stabilizing bar can be interpreted as a dynamic feedback system for the roll and pitch rates. The

system consists of a so-called flybar (a teetering rotor placed at a 90° rotation interval from the main rotor blades and tipped on both ends by aerodynamic paddles) and a mixing device that combines the flybar flapping motion with the cyclic inputs to determine the cyclic pitch angle applied to the main rotor blades.

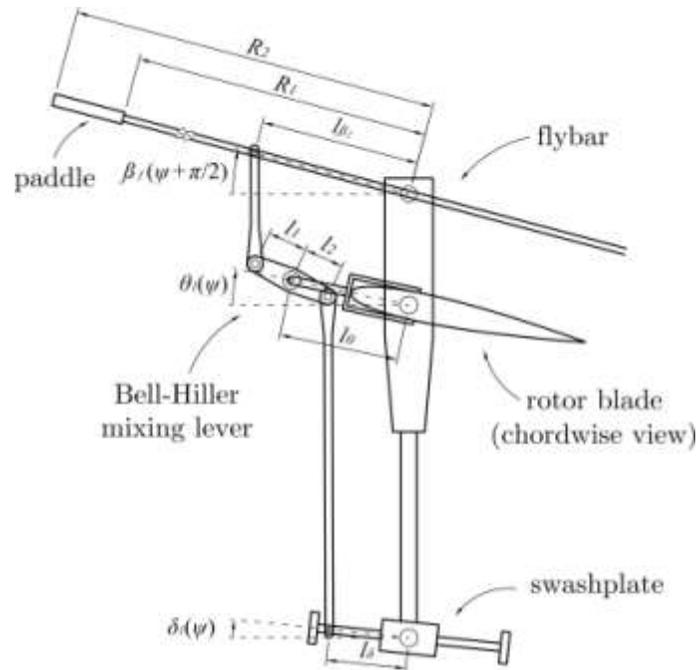


Figure 3.VI.8: Bell-Hiller system with angular displacements.

The system derives from a combination of the Bell stabilizing bar, fitted with a mechanical damper and weights at each tip, and the Hiller stabilizing bar, which instead of weights uses small airfoils with incidence commanded by the cyclic inputs. In the Hiller system, the blade pitch angle is determined by the flybar flapping only. The Bell-Hiller system introduces the mixing device that allows some of the swashplate input to be directly applied to the blades.

The flybar and main rotor flapping motions are governed by the same effects, namely the gyroscopic moments due the helicopter roll and pitch rates. However, unlike the main rotor, the flybar is not responsible for providing lift or maneuvering ability. Thus, it can be designed to have a slower response and provide the desired stabilization effect.

The notation used to describe the Bell-Hiller system is presented in figure 3.VI.8, where the mechanical arrangement is reproduced.

Gravitation Model

The six degree-of-freedom rigid body dynamic equations could be solved under two cases, 1) Flat earth approximation and 2) WGS-84 earth model. In the flat earth approximation, no information is needed

on the home latitude and longitude. In this case, it is only needed the starting position. On the other hand, WGS-84 model needs latitude, longitude and altitude, which ECEF is generated from this (see table 3.VI.2).

Generic starting position	
$B[E_1^b E_2^b E_3^b]$	Starting body position (ft)
$uvw[u v w]$	Starting body translational velocity (ft/s)
$pqr[P Q R]$	Starting body angular rates (rad/s)
$THETA[\phi \theta \psi]$	Starting body attitude (rad)
For the WGS-84 model	
Latitude	Starting latitude (rad)
Longitude	Starting longitude (rad)
Altitude	Starting altitude (ft)

Table 3.VI.2: The starting position for both modes.
1) Flat earth approximation and 2) WGS84 earth model.

In both cases, the six degree-of-freedom rigid body equations need to be initialized with the desired starting values. The gravity and gravitational model, it is based on the WGS-84 ellipsoid model of the Earth (see table 3.VI.3 and 3.VI.4).

Input	
Latitude	Latitude (rad)
Altitude	Altitude (ft)
Output	
\vec{g}	Local gravity vector in earth frame [gN gE gD] (ft/s ²)
\vec{G}	Local gravitation vector in earth frame [GN GE GD] (ft/s ²)

Table 3.VI.3: Gravity model parameters.

$WGS84.a = 6378137.0$	WGS-84 semimajor axis (m)
$WGS84.b = 6356752.3142$	WGS-84 semiminor axis (m)
GM_{\oplus}	$GM_{\oplus} = 3.9860015 \times 10^{14} \text{ (m}^3/\text{s}^2)$
$f = (WGS84.a - WGS84.b)/WGS84.a$	Flattening of the Earth
$e = \sqrt{2f - f^2}$	Eccentricity of the Earth
ψ	Geocentric latitude, (rad)
$C_2 = -0.48416685 \times 10^{-3}$	
$P_2 = \sqrt{5} / 2 (3 \sin(\psi)^2 - 1)$	
$J_2 = -\sqrt{5} C_2$	
$w_e = 2 \pi / (24 \ 60 \ 60),$	rad/s
R_e	Distance from you to center of the Earth (m)
$N \equiv WGS84.a / \sqrt{1 - e^2 \sin^2(\psi)},$	

Table 3.VI.4: Ellipsoidal earth model.

Physical Parameters

The helicopter parameters are summarize in the table 3.VI.5 and figure 3.VI.9.

Rigid Body and Fuselage	
m	mass (slugs)
I_{xx}, I_{yy}, I_{zz}	moments of inertia (slug-ft ²)
I_{xz}	cross-moment of inertia (slug-ft ²)
S_x, S_y, S_z	aerodynamic reference areas (ft ²)
V_m, V_n	aerodynamic reference volumes (ft ³)
HP_{lost}	HP lost in transmission
Generic Rotor	
n_b	number of blades
Ω	rotor speed (rad/s)
R	rotor disk radius (ft)
c	blade chord (ft)
a_0, c_{l0}	lift curve slope (1/rad) and offset
c_{d0}, c_{d1}, c_{d2}	lift dependent profile drag coefficients
I_β	flap moment of inertia (slug-ft ²)
k_β	flapp spring stiffness(N m /rad)
ρ	air density (slug/ft ³)
$\gamma = \frac{\rho c a_0 R^4}{I_\beta}$	Lock number
$S_\beta = \frac{8}{\gamma} \frac{k_\beta}{I_\beta \Omega^2}$	Siftness number
Main Rotor Specific	
γ_s	fixed shaft pitch angle (rad)
l_{mr}, h_{mr}	distances of main rotor hub aft and above center of mass (ft)
θ_{tw}	blade linear twist (rad)
Tail Rotor Specific	
l_{tr}, h_{tr}	distances of tail rotor hub aft and above center of mass (ft)
g_t	rotor speed gearing constant
k_3	pitch/flap coupling factor
Bell-Hiller/Hiller Stabilizing Bar	
R_1, R_2	paddle starting and ending radii (ft)
$\gamma_f = \rho c_f a_{0f} \frac{R_2^4 - R_1^4}{I_{\beta f}}$	swashplate and flybar linkage ratios
Empennage	
l_{tp}, h_{tp}	distances of horizontal tailplane aft and above center of mass (ft)
l_{fn}, h_{fn}	distances of vertical fin aft and above center of mass (ft)
a_{0tp}, a_{0fn}	lift curve slopes (1/rad)
S_{tp}, S_{fn}	areas (ft ²)
α_{0tp}	tailplane zero-lift incidence angle (rad)
β_{0fn}	fin zero-lift sideslip angle (rad)

Table 3.VI.5: Helicopter Parameters

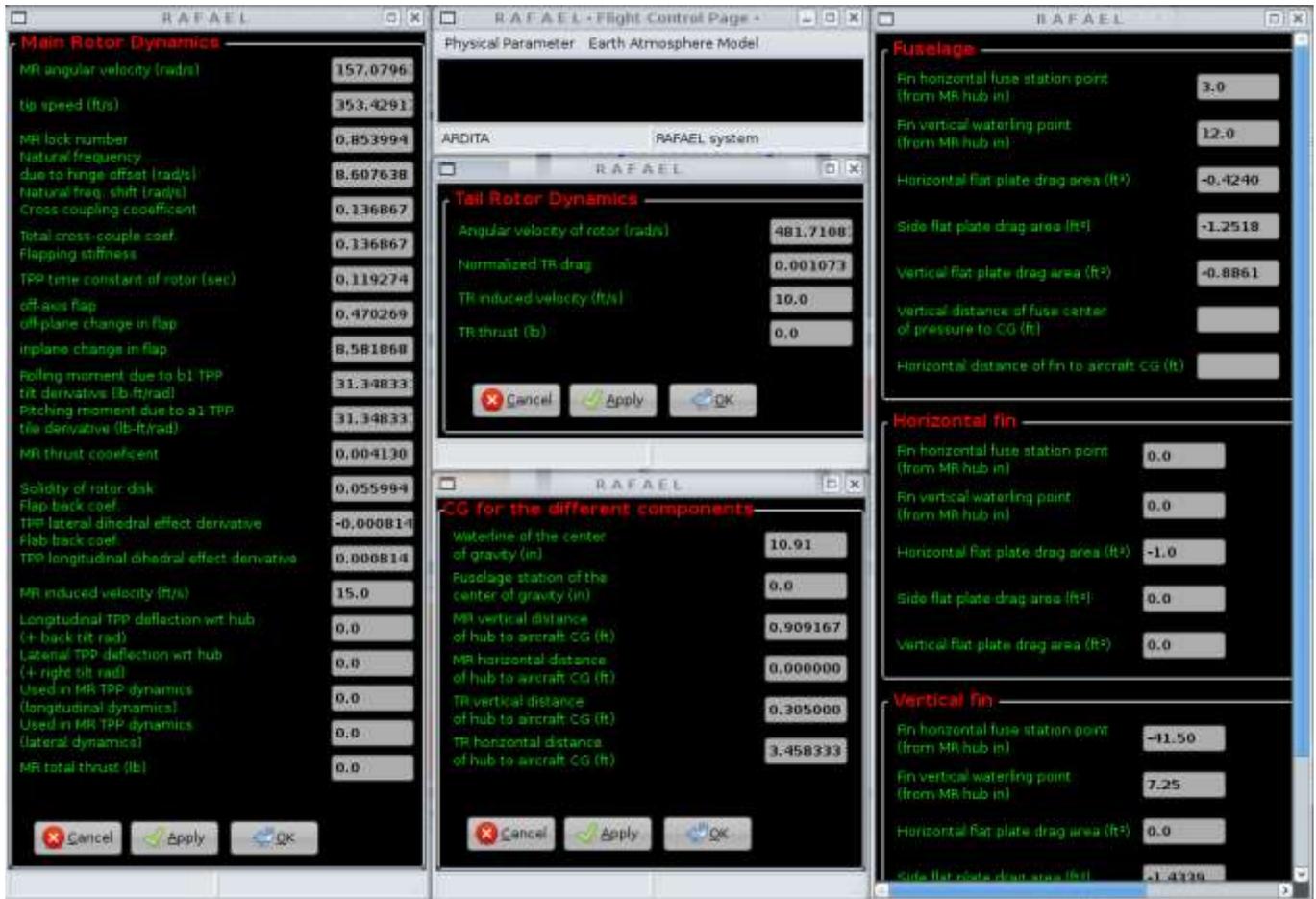


Figure 3.VI.9: Helicopter parameters shown by *Raphael* GCS.

Wind model

We have adopted a simple wind model. It is a dynamic system that will generate random winds, up to a maximum value, in both vertical and horizontal directions. Each wind component is modeled by the following transfer function in fixed-body frame:

$$\frac{y}{u} = \frac{S + \omega_n^2}{S^2 + 2\zeta\omega_n S + \omega_n^2}$$

where, $\omega_n = 1.5$ and $\zeta = 0.00005$.

Servo control input

In this section, we will provide a servo-actuator model. It is a second-order dynamic model which is tunable via ω_n and ζ values. The transfer function model of the servo used is as follows:

$$\frac{y}{u} = \frac{S + \omega_n^2}{S^2 + 2\zeta\omega_n S + \omega_n^2}$$

We have taken the Futaba 9202 servo as our sample-servo model, with parameters given by: $\omega_n = 38.2261$ and $\zeta = 0.5118$. However, the digital Futaba 9253 servo is much faster than Futaba 9202 servo. The parameters are given by: $\omega_n = 32.2261$ and $\zeta = 0.5118$. The propagation state is done using RK4 routine.

Atmosphere Earth Model

In this section, we define an atmospheric model which yields the density air, air pressure, air temperature, local speed of sound as a function of current density altitude. The table 3.VI.6 shows the atmospheric model parameters.

ρ	Density of air (slug/ft ³)
pressure, P_{air}	Air pressure (lb/ft ²)
temperature, T_{air}	Air temperature (°R)
sp_sound, c_s	Local speed of sound (ft/s)
densAlt(altitude H)	Current density altitude (ft MSL + up)

$$\begin{aligned} \rho &= 0.0023769(1 - (0.68753 - 0.003264 H \times 10^{-5}) H \times 10^{-5})^{4.256} \\ T_{air} &= 518.69(1 - 0.687532 H \times 10^{-5} + 0.003298(H \times 10^{-5})^2) \\ P_{air} &= 1716.5 \rho T_{air} \\ c_s &= 1116.45 \sqrt{T_{air}/518.69} \end{aligned}$$

Table 3.VI.6: Atmosphere Earth model

3.VI PHOTOGRAMMETRY MODULE

This is a complete photogrammetry tool, consisting of tools such as normalization, triangulation, stereo-plotter, rectification, interior orientation, exterior orientation and DEM modelling. Unlike other similar tools though, it also has a 6-parameter iterative algorithm capable of correcting imaging system distortions, so mosaic generation and image registration improves, particularly where images are sourced from multiple disparate platforms are being fused together. This module needs to be upgraded for use by the MOD by enabling the calibration procedure to use known targets such as a helicopter or aircraft, instead of the usual dot patterns as shown below. In this way as a UAV takes off, its imaging system can take a few frames of a known target and later these will be used to calibrate the imaging system and correct for distortion.

In order to do the more advanced tasks, such as cartography, ortho-rectification, target-acquisition, DEM modeling, mosaicing, etc. high quality images must be produced from UAV-derived imaging, which requires modules such as a 6-degree pixel calibration camera correction tool, capable of correcting for camera and lens distortion. Other functions available in this module include normalization, photo-triangulation, stereo-plotter (anaglyph), rectification, interior and exterior orientation correction.

In summary, there are functions for:

- Camera Calibration
- Rectification
- Interior orientation
- Exterior orientation
- Photo-triangulation
- Stereo-plotter (anaglyph)

Camera Calibration

Numerical example for camera calibration module: In order to calibrate any camera, we have used a chessboard pattern of known dimensions. Each black/white square on the chessboard has 3 cm x 3 cm of size. Thus, we have used 10 images with 640 x 512 pixel size. These images were taken from distinct angular directions. The calibration algorithm given by Heikkila (1997) used on a standard CCD camera gives the following results:

Parameter	Value
Principal Point	(321.9588, 237.3126) pixel
Scale Factor	0.9989
Effective focal length	3.2864 mm
Radial distortion	$k_1 = 2.422831 \times 10^{-3} \text{ mm}^{-2}$ $k_2 = -1.234360 \times 10^{-3} \text{ mm}^{-4}$
Tangential distortion	$p_1 = 1.343740 \times 10^{-3} \text{ mm}^{-1}$ $p_2 = -7.714951 \times 10^{-4} \text{ mm}^{-1}$

Table 3.VI.1: It shows calibration parameter for a typical digital camera.

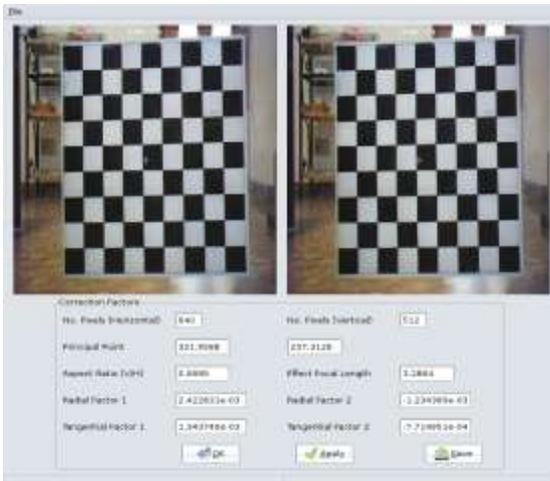


Figure 3.VI.1: A window of the camera calibration module.

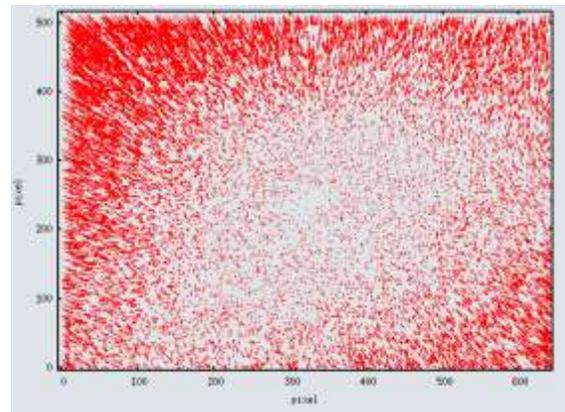


Figure 3.VI.2: The radial and tangential effects are shown. The effect has been exaggerated 10 times.

Orthorectification

Cartography: having properly corrected source UAV images enables a successful and highly accurate ortho-rectification and tiling process. Direct Linear Transformation, Projective, 2D Polygonal, 3D Polygonal, Linear General Affine, Non-Linear Rigid Body, Non-linear Orthogonal and Linear Isogonal. Figure 3.VI.3 shows a typical orthorectification for a pair of images.



View A



View B



Figure 3.VI.3: View A shows a image taken with a certain camera pose, while View B shows another view taken with different camera pose. The ortho-rectified image is shown too.

One of the challenges of full-motion video exploitation lies in how to present the images to the user in such a way as to maximize comprehension of the information. One of the effects that minimizes comprehension is having only a localized view of the video information presented in each frame. To

reduce this effect, a portion of video can be viewed as a mosaic of images. This birds-eye view increases spatial comprehension of the data allowing more accurate analysis of the information. This module is currently being added and will include super-resolution, increased depth-of-field, blurring correction as required.

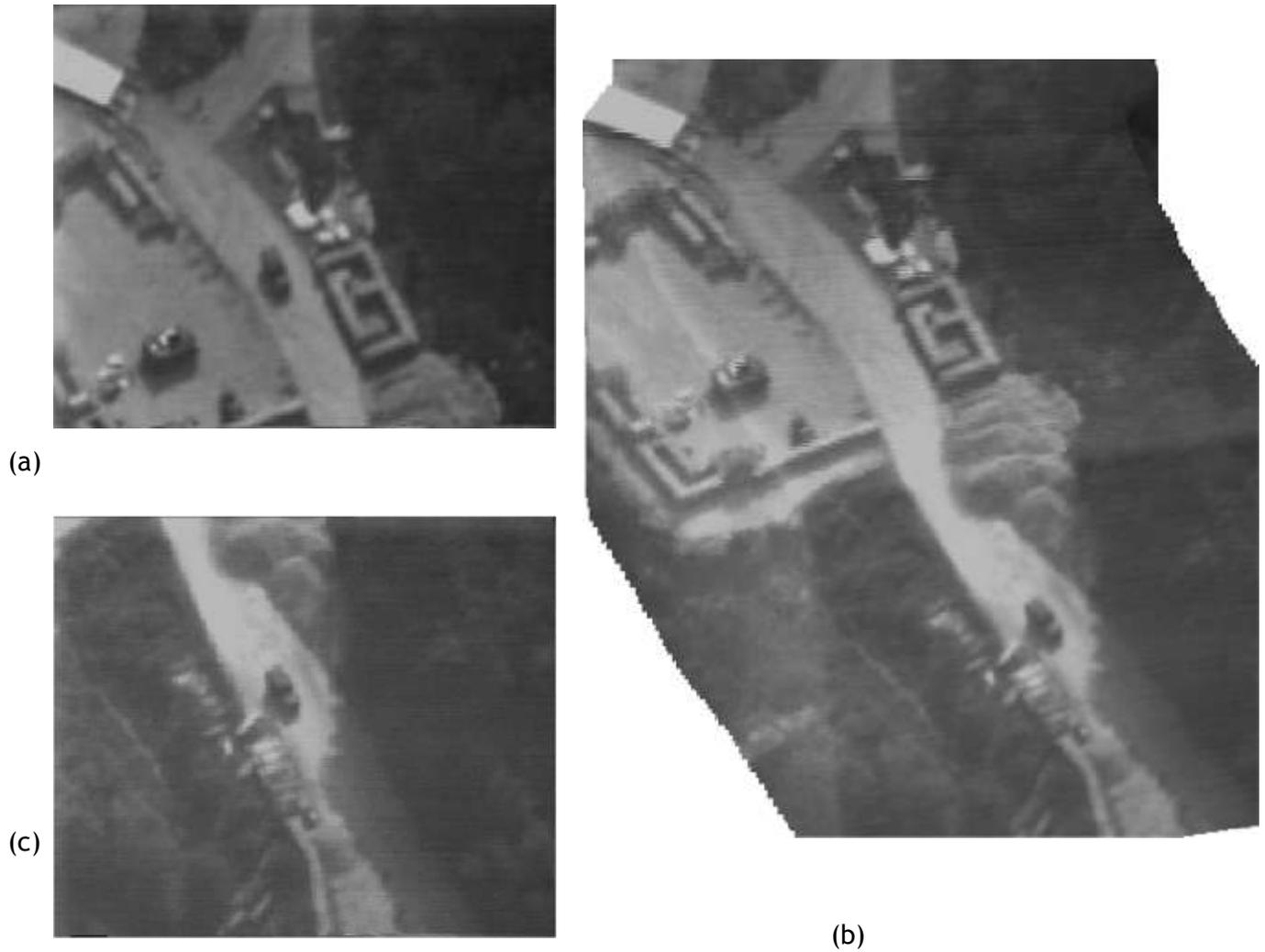


Figure 3.VI.4. Images (a) and (b) show the first and 180th frames of the Predator F sequence. The vehicle near the center moves as the camera pans across the scene in the same general direction. Poor contrast is evident in the top right of (a), and in most of (b). The use of basis functions for computing optical flow pools together information across large areas of the sequence, thereby mitigating the effect of poor contrast. Likewise, the iterative process for obtaining model parameters successfully eliminates outliers caused by the moving vehicle. The mosaic constructed from this sequence is shown in (c).

3.VII IMAGE RESTORATION

Our sophisticated photogrammetry module allows us to make a number of improvements to restore images, unavailable in other products, such as automatic blurring removal and another common problem with UAV-derived imaging: that we often have 5-10 low-quality images of a target and require a single high-resolution image. So, an advanced sub-pixel super-resolution resolution algorithm using wavelets was implemented.

Super-resolution: Another common problem with UAV-derived imaging is that we often have 5-10 low-quality images of a target and require a single high-resolution image. So, an advanced sub-pixel resolution algorithm using wavelets was implemented.

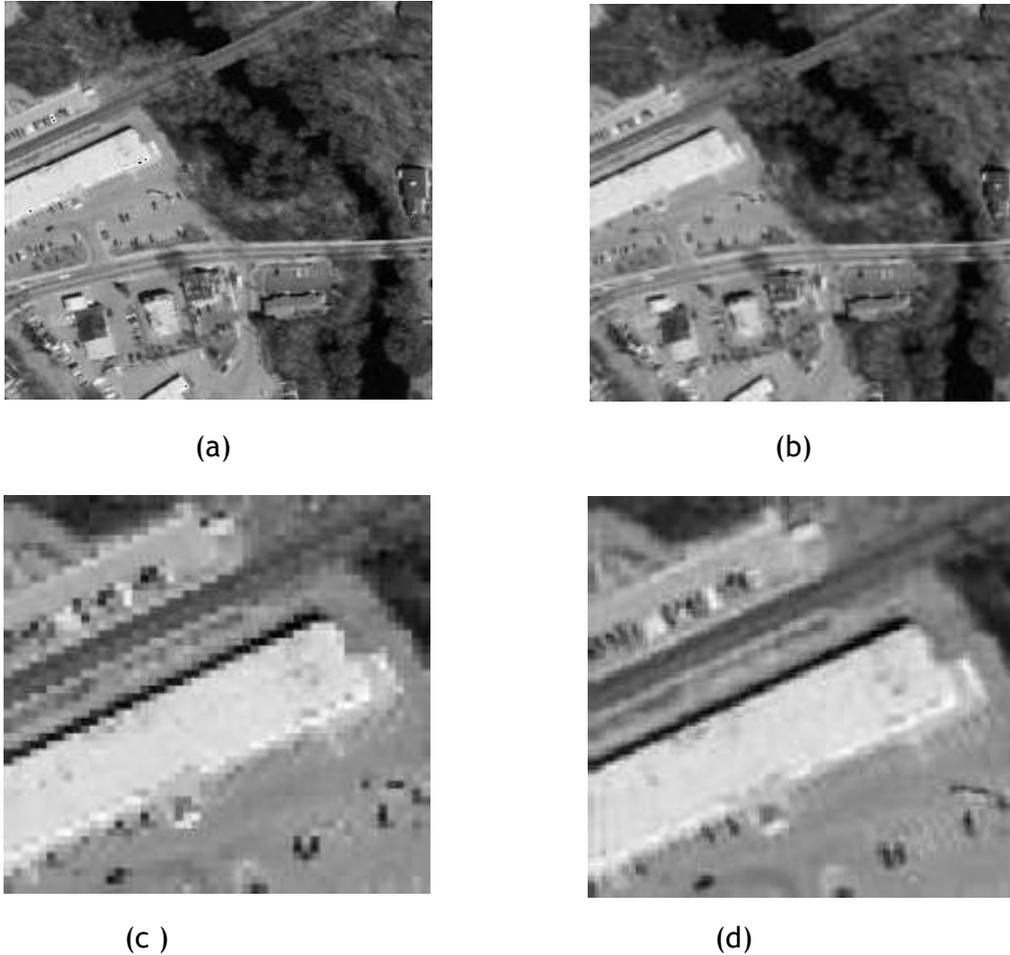
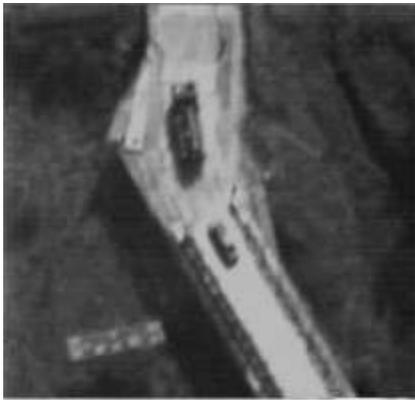


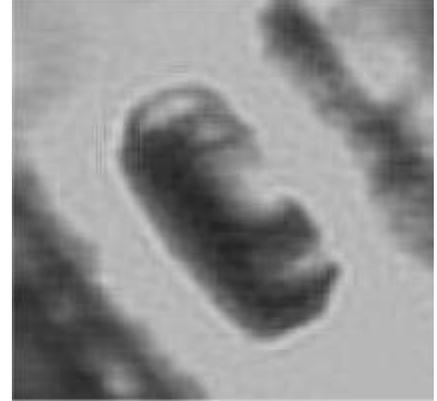
Figure 3.VII.1: Example of our super-resolution algorithm. (a) Original image, (b) Super-resolution image. (c) Single image zoom and (d) Super-resolution detail.



(a)



(b)



(c)



(d)



(e)



(f)

Figure 3.VII.2: A demonstration for performing motion super-resolution is presented here. The Predator B sequence data is gathered from an aerial platform (the predator unmanned air vehicle), and compressed with loss. One frame of this sequence is shown in (a). Forty images of this sequence are co-registered using an affine global motion model, upsampled by a factor of 4, combined and sharpened to generate the super-resolved image. (b) and (d) show the car and truck present in the scene, at the original resolution, while (e) shows the truck image upsampled by a factor of 4, using a bilinear interpolator. The super-resolved images of the car and truck are shown in (c) and (f) respectively. The significant improvement in visual quality is evident.

Increase Depth of Field

Digitally-increased depth of field: Where ambient conditions require a low $f/\#$ sometimes only a portion of the image is in focus, so we developed a digital means to combine a series of images with a varying focal plane so as to obtain a single in-focus image throughout.

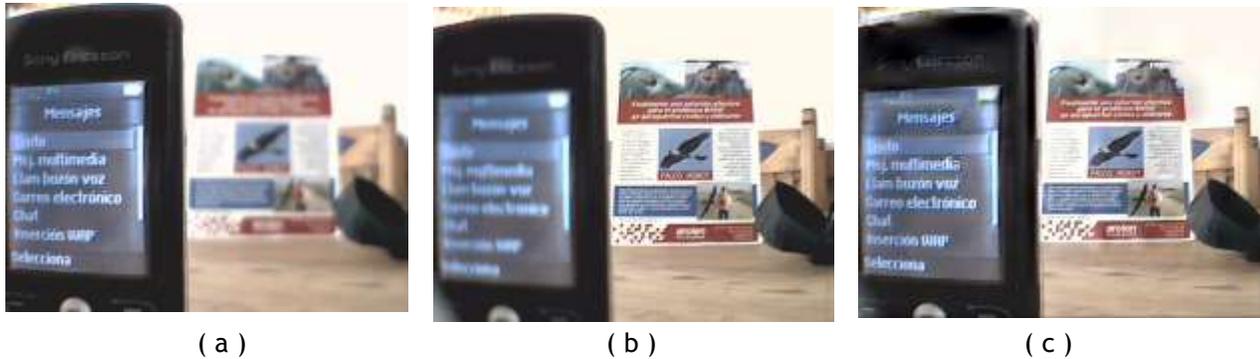


Figure 3.VII.3: (a) Focused image on first plane, (b) focused image on second plane and (c) increased depth-of-field image.

Blurring: Another common problem with UAV-derived imaging is that due to vibration making images blurred very often. So, an advanced automatic blurring correction tool has been implemented.

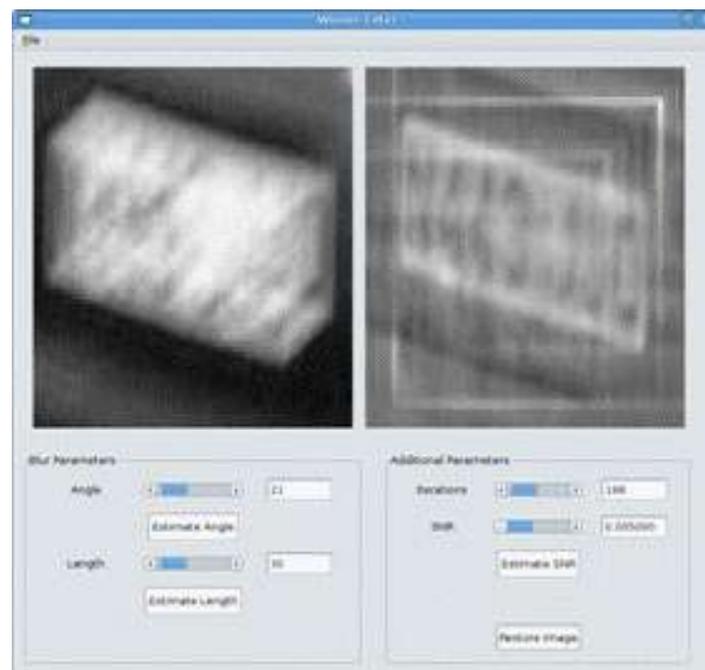


Figure 3.VII.4: Automatic de-blurring

3.VIII

RAPHAEL GEOGRAPHICAL INFORMATION SYSTEM

A problem often encountered by users is that the quality of unmanned vehicle images is simply not of sufficient quality to analyze the data in a GIS. Moreover, the interface between UAV-derived images and a GIS is extremely difficult due to the image quality and referencing requirements of a GIS. The image quality restoration available in **Raphael GCS** makes it possible to combine UAV aerial imagery with existing data such as geological, topological, land use, rain fall statistics, etc. The GIS module currently has the following basic functions:

- Analysis of both vector and raster images
- User-friendly
- Import/export of popular data formats
- On-screen and tablet digitizing
- Comprehensive set of image processing tools
- Ortho-photo
- Cartography
- Image geo-referencing and registration
- Transformation/rectification and mosaicing
- Advanced modeling and spatial data analysis
- Rich projection and coordinate system library
- Geo-statistical analysis, with Kriging for improved interpolation
- Production and visualization of stereo image pairs
- Spatial multiple criteria evaluation

In principle, data in the GIS module is organized according to its nature, five different types of data objects can be addressed: tables, shapes (vector data), TIN (Triangular Irregular Network), Point Cloud, and grid (raster data). All data object classes derive from the same base class, which defines basic properties (name, description, data history, file path) and methods (load, save, copy, clear). Each of the derived classes has specific methods for data access and manipulation.

Tables: Table objects are collections of data records, which hold the data itself in data fields. Data fields can be of different type, e.g. character string or numerical value. Tables are a very powerful tool for displaying data. Tables are important because they constitute a linking element between **Raphael GCS** and other application such as spreadsheets. Not all **Raphael GCS** modules or functions generate new grids as result. Some of them return tabular data instead of new grids, and you should know how to handle this kind of result. Also, tables are required sometimes as input. File access is supported for text and DBase formats.

Shapes: While grid (raster data) is contained in the grid itself, vector data objects needs to be stored in a database. This causes a vector layer to be separated in two different entities: the “shapes” (points, lines, polygons) where the spatial information is kept, and the database where the information about those shapes is stored. There are many modules for manipulation and analysis of vector data, like merging of layers, querying of shapes, attribute table manipulation, type conversion and automated document creation. Standard operations on vector data are polygon layer intersections and vector data creation from raster data, e.g. of contour lines. The built-in file format is the ESRI Shape File format (ESRI 1998).

TIN: It is a special vector data structure for point data, for which the neighborhood relations of the points are defined by Triangular Irregular Network (TIN) using Delaunay Triangulation. Similar to shapes, TIN has associated table object for the storage of additional attribute data. TIN can be loaded and saved as points in the ESRI Shape File format.

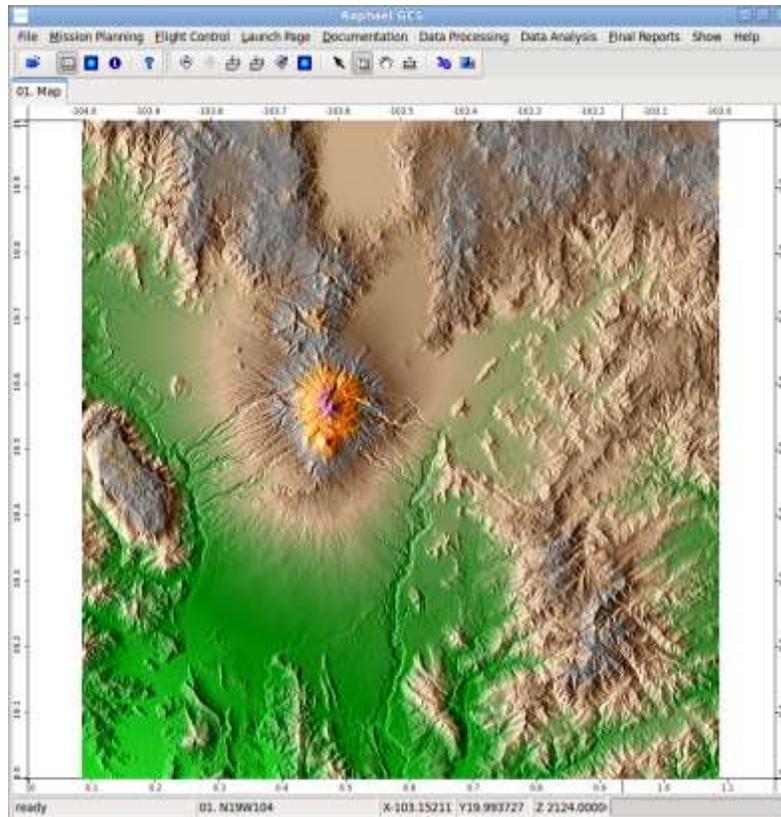


Figure 3.VIII.1: It is shows Digital Elevation Model of the Colima Volcano in Jalisco, Mexico.

Grids: Raster or grid data objects are matrix stored in numerical values. Possible data types are 1 bit, 1, 2, 4 byte integers, 4 and 8 byte floating-point values. *Raphael GCS* contains standard tools for grid manipulation, for example: Grid calculator, where a user-defined formula is used to combine an arbitrary number of raster layers. The raster data access methods supported by *Raphael GCS* have plenty of import/export raster format, which uses information on how to interpret the actual data file. Raster data can be created from point data using nearest neighbour, triangulation and other interpolation techniques. Modules for the construction and preparation of raster data, allow the resampling, closing of gaps and the value manipulation with user defined rules. A number of filter algorithms exist for smoothing, sharpening or edge detection. Classifications can be performed using cluster analysis or a supervised procedure, like Maximum Likelihood classification. Data analyses cover image, pattern and cost analysis. Other standard operations are skeletonisations and bufferings. However, the fundamental task to do is the visualization of raster data. *Raphael GCS* is capable of visualize raster data easily. A typical Digital Elevation Model is shown in the figure 3.VIII.1. The image has great level of detail, identifying high area and low areas.

The combination of shapes (vector data) and grid (raster data) has many advantages, and *Raphael GCS* is perfectly capable of combining them and making it easy for the user to handle both data types without effort. For example, if you want to create a DEM from point data by interpolating the height information associated with those points, *Raphael GCS* will have no problem. For a detailed list of all available functions, see Appendix 8.1.

However, final users such as aid agencies, search-and-rescue, agriculture planning, disaster relief units, etc. often require more sophisticated reports. In the following section we present advanced modules already implemented on *Raphael GCS*.

Simulation - Cellular Automata -

Conway's Life: It would be incomplete to explain without an example. The history of cellular automata dates back to the forties with Stanislas Ulam. This mathematician was interested in the evolution of graphic constructions generated by simple rules. The base of his construction was a two-dimensional space divided into "cells", a sort of grid. Each of these cells could have two states: ON or OFF. Starting from a given pattern, the following generation was determined according to neighbourhood rules. For example, if a cell was in contact with two "ON" cells, it would switch on too; otherwise it would switch off. Ulam, who used one of the first computers, quickly noticed that this mechanism permitted to generate complex and graceful figures and that these figures could, in some cases, self-reproduce. Extremely simple rules permitted to build very complex patterns.

There are three fundamental properties of cellular automata:

- *Parallelism:* A system is said to be parallel when its constituents evolve simultaneously and independently. In that case cells update are performed independently of each other.
- *Locality:* The new state of a cell only depends on its actual state and on the neighbourhood.
- *Homogeneity:* The laws are universal, that's to say common to the whole space of cellular automata.

The rules are quite simple:

1. One inactive cell surrounded by three active cells becomes active ("it's born")
2. One active cell surrounded by 2 or 3 active cells remains active
3. In any other case, the cell "dies" or remains inactive.

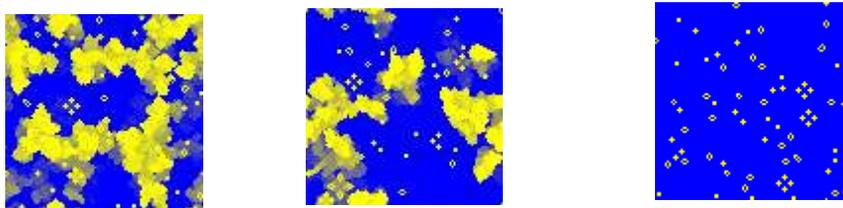


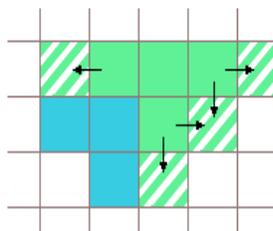
Figure 3.IX.1: Simulation of Conway's Life cellular automata.

Wa-Tor: An ecological simulation of predator-prey populations. It simulates the hypothetical toroidal Planet Wa-Tor (Water Torus) whose surface is completely covered with water, occupied by two species: fish and sharks. The sharks are the predators. They eat the fish. The fish exist on a never ending supply of plankton. Both sharks and fish live according to a strict set of rules. This simulation of a simple ecology is highly dynamic as both species are walking a thin line between continuing life and extinction.

Rules: In general, the simulation is based on discrete time steps. The simulation runs on a rectangular grid. To represent the toroidal world, opposing sides of the grid are connected. If an individual moves out on one side of the simulation domain, it reenters immediately on the opposing side. Fish and shark move every time step (if possible) and interact according to the following set of rules.

Rules for fish: In every time step, a fish moves randomly to one of the four neighboring fields, provided it is empty. Every fish has a predefined "breed time". On exceeding this time, it gives birth to a new fish in one of the neighboring cells, provided this randomly selected cell is free. (If not nothing happens.) Breed time counter of both the original and the descendant fish will be reset to zero. Technically fish never die. They live until they reach the breed time, then they clone and both parent as well as offspring restart their life cycle. The following picture shows options for prey movement. Arrows indicate possible movements. Fish are not allowed to move to cells occupied by sharks. If there are no free neighboring cells no movement occurs.

Fish movement

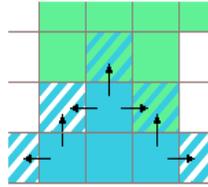


- ➡ create offspring in free neighboring cells (▨)
- ➡ only move to free cells (↑→)

Figure 3.IX.2: Fish movement for Wa-tor cellular automata.

Rules for sharks: Sharks move randomly to fields that are either free or occupied by fish. Every round they lose one point of energy. If they enter a field occupied by a fish they eat the fish and gain a defined amount of energy. If the energy level drops below zero the shark dies. If the energy exceeds a predefined value sharks create an offspring in a free neighboring field. The energy is split evenly between the parent and the child.

Shark movement



- ➡ only move to free or fish cells (↑→)
- ➡ when moving to fish cells eat fish and increase energy (■)
- ➡ if enough energy is accumulated create offspring (■) in free neighboring cells
- ➡ every round decrease energy by one point
- ➡ die if energy drops below zero

Figure 3.IX.3: Shark movement for Wa-tor cellular automata.

The figure 3.IX.4 below shows the simulation of a Wa-tor Automata given by *Raphael GCS*.

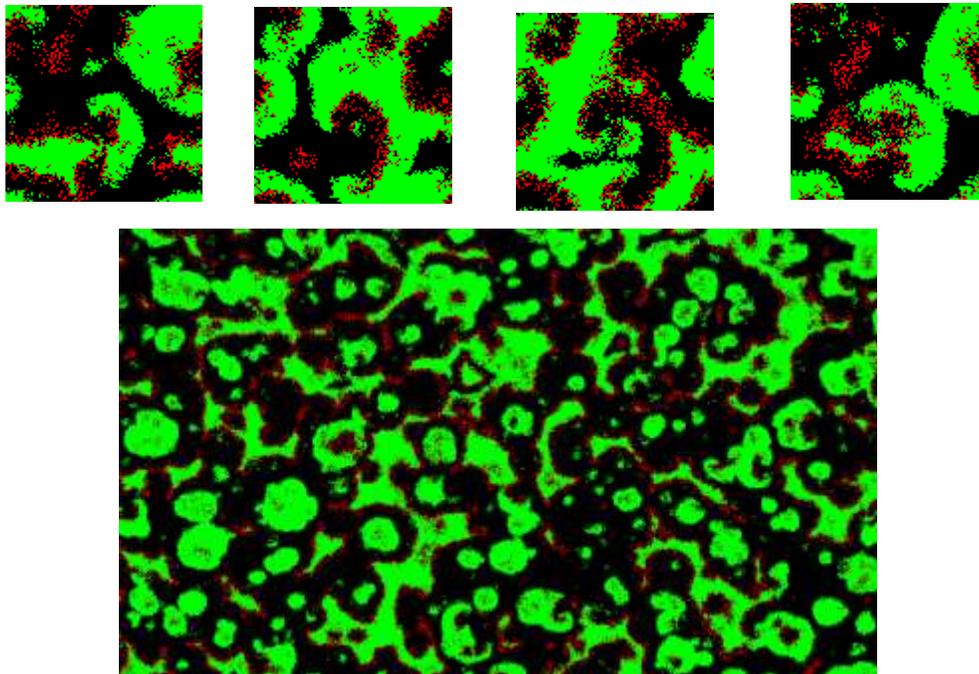


Figure 3.IX.4: The wa-tor cellular automata.

Simulation - Fire Risk Analysis -

Fire Risk Analysis: This module predicts danger, compound probability and priority index for a given Digital Elevation Model (DEM), fuel model, grid set of fuel moisture, wind speed and its direction. It is based on the BEHAVE fire modelling system supported by the U.S. Forest Service, Fire and Aviation Management, (see at <http://fire.org>).

Input Data – Name	Type	Label	Description
DEM	Grid	DEM	Digital Elevation Model
Fuel Model	Grid	FUEL	
Wind Speed	Grid	WINDSPD	Wind speed (m/s)
Wind Direction	Grid	WINDDIR	Wind direction (degrees clockwise from north)
Dead Fuel Moisture 1H, 10H, 100H	Grid	M1H, M10H, M100H	
Herbaceous Fuel Moisture	Grid	MHERB	
Wood Fuel Moisture	Grid	MWOOD	
Value	Grid (optional)	VALUE	
Base Probability	Grid (optional)	BASEPROB	

Table 3.IX.1: Fire Risk Analysis, input data.

Output data - Name	Type	Label
Danger	Grid	DANGER
Compound Probability	Grid	COMPPROB
Priority Index	Grid	PRIORITY

Table 3.IX.2: Fire Risk Analysis, output data.

Simulation - Fire Spreading Analysis -

Fire spread analysis: This module predicts the spread rate, intensity, flame length and scorch height of free-burning surface fires for a given Digital Elevation Model (DEM), fuel model, grid set of fuel moisture, wind speed and its direction. It is based on the BEHAVE fire modelling system supported by the U.S. Forest Service, Fire and Aviation Management, (see at <http://fire.org>).

Input Data - Name	Type	Label	Description
DEM	Grid	DEM	Digital Elevation Model
Fuel Model	Grid	FUEL	
Wind Speed	Grid	WINDSPD	Wind speed (m/s)
Wind Direction	Grid	WINDDIR	Wind direction (degrees clockwise from north)
Dead Fuel Moisture 1H, 10H, 100H	Grid	M1H, M10H, M100H	
Herbaceous Fuel Moisture	Grid	MHERB	
Wood Fuel Moisture	Grid	MWOOD	
Ignition Points	Grid	VALUE	

Table 3.IX.3: Fire Spreading Analysis, input data.

Output data - Name	Type	Label	Description
Time	Grid	TIME	Time since initial ignition.
Flame Length	Grid	FLAME	Flame length (m)
Intensity	Grid	INTENSITY	Intensity (Kcal/m)

Table 3.IX.4: Fire Spreading Analysis, output data.

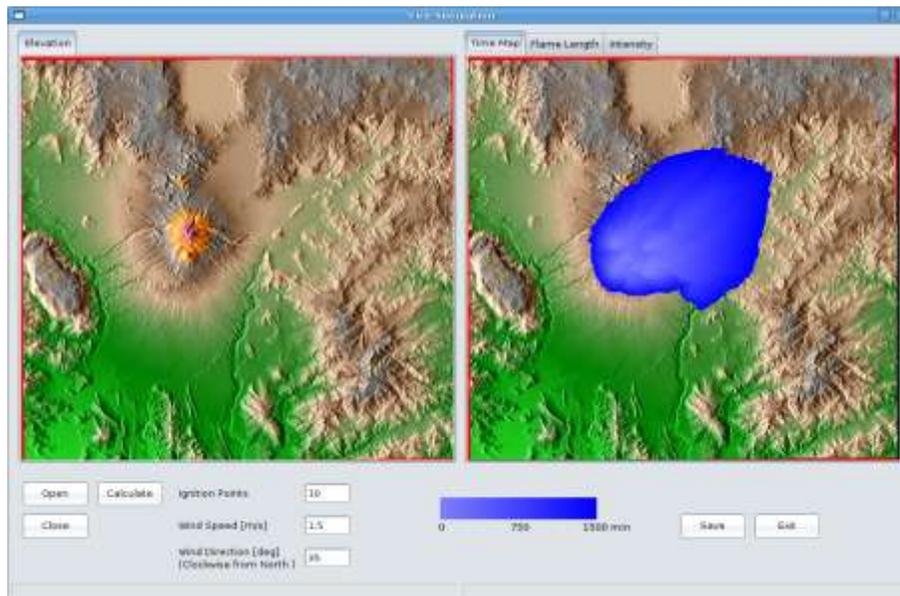


Figure 3.IX.5: Time spread (mins)

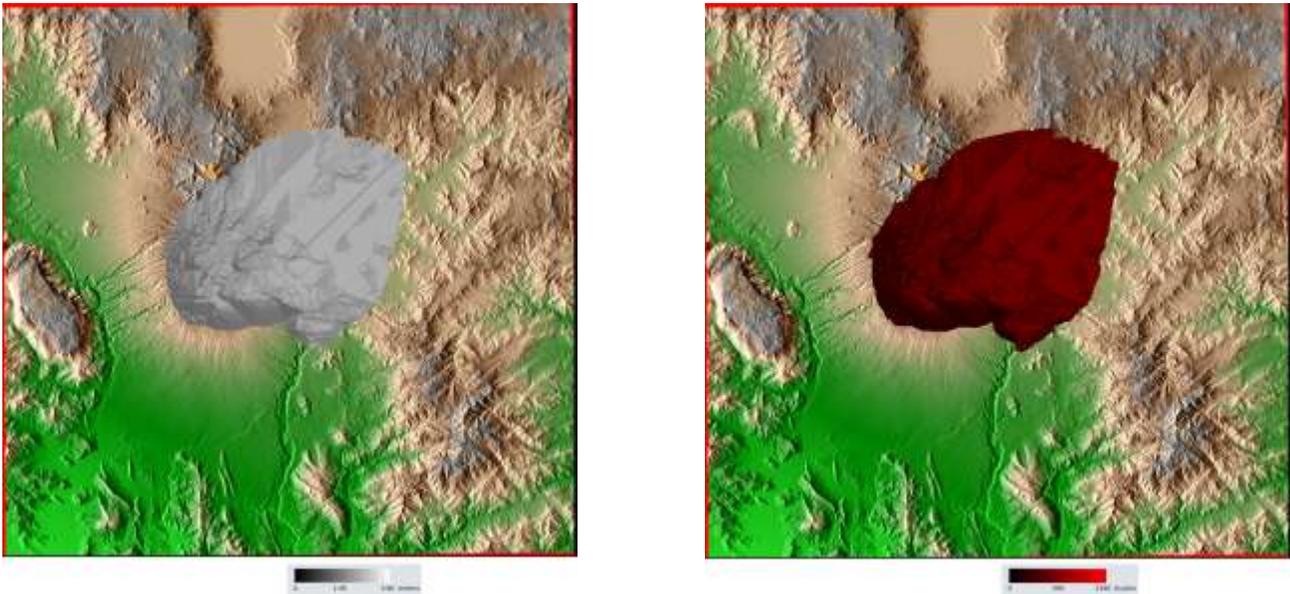


Figure 3.IX.6: Left) Flame Length (m), Right) Intensity (Kcal/m)

Simulation - Hydrology -

Overland Flow (Kinematic Wave D8): A distributed hydrologic model, known as the Terrestrial Hydrologic Model or THM was developed for use with rasterized databases to simulate surface runoff. Computations are performed on a pixel-by-pixel basis and all physical drainage basin properties including area, slope, stream length, and stream order are obtained or estimated from a digital elevation model (DEM). Other data sets, such as curve numbers or infiltration rates, are required for estimating the hydrologic abstractions. Precipitation is supplied in the form of gage input, uniform distributions, or raster data. At the present time, hydrologic abstractions can be estimated by any of three methods: a constant infiltration rate, the Soil Conservation Service curve number method, or solution of the more physically based Green-Ampt equation. Overland flow is computed by a kinematic wave approximation and channel routing is performed using the Muskingum-Cunge method.

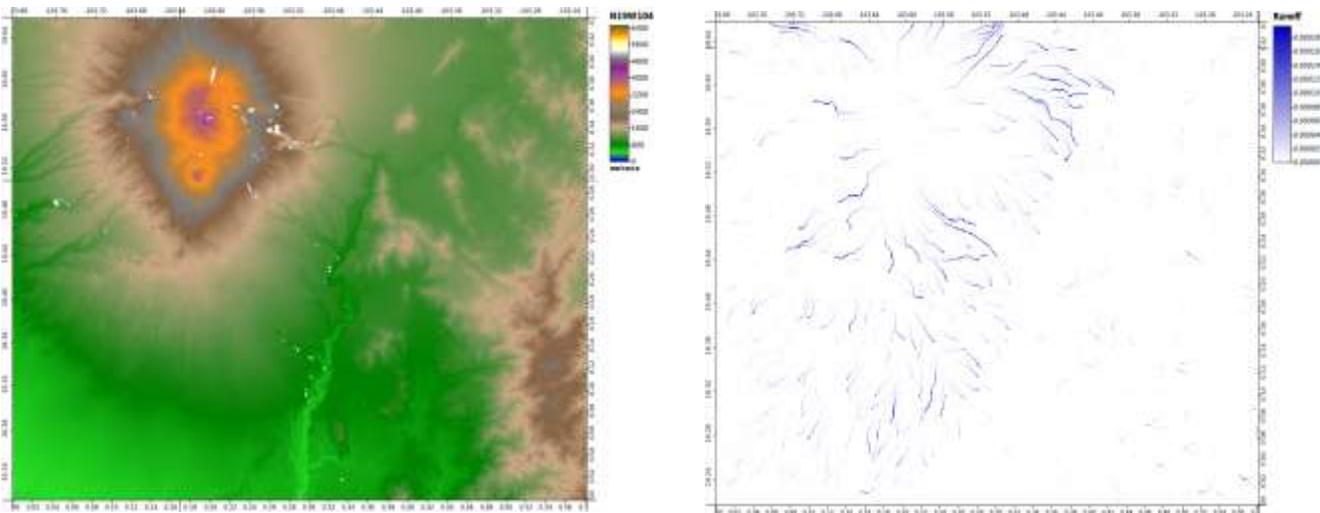


Figure 3.IX.7: It shows the Runoff of a DEM.

Soil Moisture Content: The WEELS (Wind Erosion on European Light Soils) soil moisture model dynamically calculates the soil moisture based on the rules proposed by the DVWK (1996) with input data about:

- Soil properties: Field capacity and permanent wilting point (grid type).
- Land use: crop types (grid type).
- Climate: Daily values of precipitation, temperature and air humidity (table type).

TOPMODEL: This Simple Subcatchment Version of TOPMODEL. Based on the 'TOPMODEL demonstration program (v95.02) by Keith Beven (Centre for Research on Environmental Systems and Statistics, Institute of Environmental and Biological Sciences, Lancaster University, Lancaster LA1 4YQ, UK) and the C translation of the Fortran source codes implemented in GRASS.

This program allows single or multiple subcatchment calculations but with single average rainfall and potential evapotranspiration inputs to the whole catchment. Subcatchment discharges are routed to the catchment outlet using a linear routing algorithm with constant main channel velocity and internal

subcatchment routing velocity. The program requires $\ln \left[\frac{a}{\tan B} \right]$ distributions for each subcatchment.

These may be calculated using the GRIDATB program which requires raster elevation data as input. It is recommended that those data should be 50 m resolution or better.

NOTE: TOPMODEL is not intended to be a traditional model package but is more a collection of concepts that can be used (where appropriate). It is up to the user to verify that the assumptions are appropriate (see discussion in Beven et al.(1994). This version of the model will be best suited to catchments with shallow soils and moderate topography which do not suffer from excessively long dry periods. Ideally predicted contributing areas should be checked against what actually happens in the catchment.

It includes infiltration excess calculations and parameters based on the exponential conductivity Green-Ampt model of Beven (HSJ, 1984) but if infiltration excess does occur it does so over whole area of a subcatchment. Spatial variability in conductivities can however be handled by specifying K_o

parameter values for different subcatchments, even if they have the same $\ln \left[\frac{a}{\tan B} \right]$ and routing parameters, i.e. to represent different parts of the area.

Note that time step calculations are explicit ie. SBAR at start of time step is used to determine contributing area. Thus with long (daily) time steps contributing area depends on initial value together with any volume filling effect of daily inputs. Also base-flow at start of time step is used to update SBAR at end of time step.

Input – Name -	Type	Label	Description
$\frac{a}{\tan B}$	Grid	ATANB	
Climate Data (P,EP)	Table	CLIMATE	

Table 3.IX.5: TOPMODEL Input data.

Optimal simulation parameters:

Name	Label	Value	Unit
Time Step	DTIME	1	h
Number of Classes	NCLASSES	30	
Initial subsurface flow per unit area	P_QS0	3.28×10^{-5}	m/h
Areal average of $\ln(T0) = \ln(Te)$	P_LNTE	5	$\ln \left[\frac{m^2}{h} \right]$
Model parameter	P_MODEL	0.032	m
Initial root zone storage deficit	P_SR0	0.002	m
Maximum root zone storage deficit	P_SRZMAX	0.05	m
Unsaturated zone time delay per unit storage deficit	P_SUZ_TD	50.0	h
Main channel routing velocity	P_VCH	3600.0	m/h
Internal subcatchment routing velocity	P_VR	3600	m/h
Surface hydraulic conductivity	P_K0	1	m/h
Wetting front suction	P_PSI	0.02	m
Water content change across the wetting front	P_DTHETA	0.1	
Green-Ampt Infiltration	BINF	TRUE	boolean

Table 3.IX.6: Initial parameter values.

Output – Name -	Type	Label	Description
Simulation Output	Table	TABLE	
Soil Moisture Deficit	Grid	MOIST	

Table 3.IX.7: TOPMODEL Output data.

Water Retention Capacity: It is another advanced hydrology module implemented on *Raphael GCS*.

Simulation - Identification of unit hydrographs and component flows from rainfall, evaporation and streamflow data (IHACRES) -

This is a catchment-scale rainfall stream-flow modeling methodology developed collaboratively by the Institute of Hydrology (IH), United Kingdom and the Centre for Resource and Environmental Studies, Australian National University (CRES at ANU), Canberra. Its purpose is to assist the hydrologist or water resources engineer to characterize the dynamic relationship between basin rainfall and stream-flow. Possible applications include:

- Identification of unit hydrographs
- Continuous time series stream-flow modeling
- Environmental change - hydrological regime studies

- Runoff event modelling
- Hydrograph separation (for example, to assist with water quality investigations)
- Derivation of a Slow Flow Index (SFI)
- Derivation of Dynamic Response Characteristics (DRCs)
- Investigation of relationships between DRCs and Physical Catchment Descriptors (PCDs)
- Teaching unit hydrograph theory and its applications
- Hydrometric data quality assurance/control (QA/QC)
- Infilling missing streamflow data

The only field-data required are time series of rainfall and streamflow, and a third variable by which evapotranspiration effects can be approximated. The third variable can be air temperature but pan evaporation, or potential evaporation (PE) derived from hydrometeorological measurements, can also be used as alternatives if they are available. Rainfall r_k is converted to effective rainfall u_k by a non-linear loss module, followed by a linear unit hydrograph (UH) module to convert effective rainfall to streamflow x_k . See figure 3.IX.8.

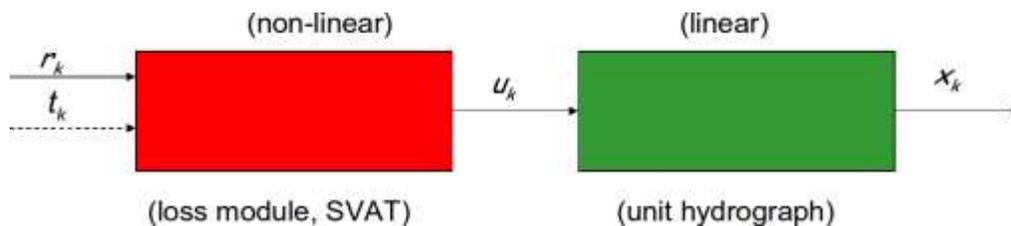


Figure 3.IX.8: IHACRES block process.

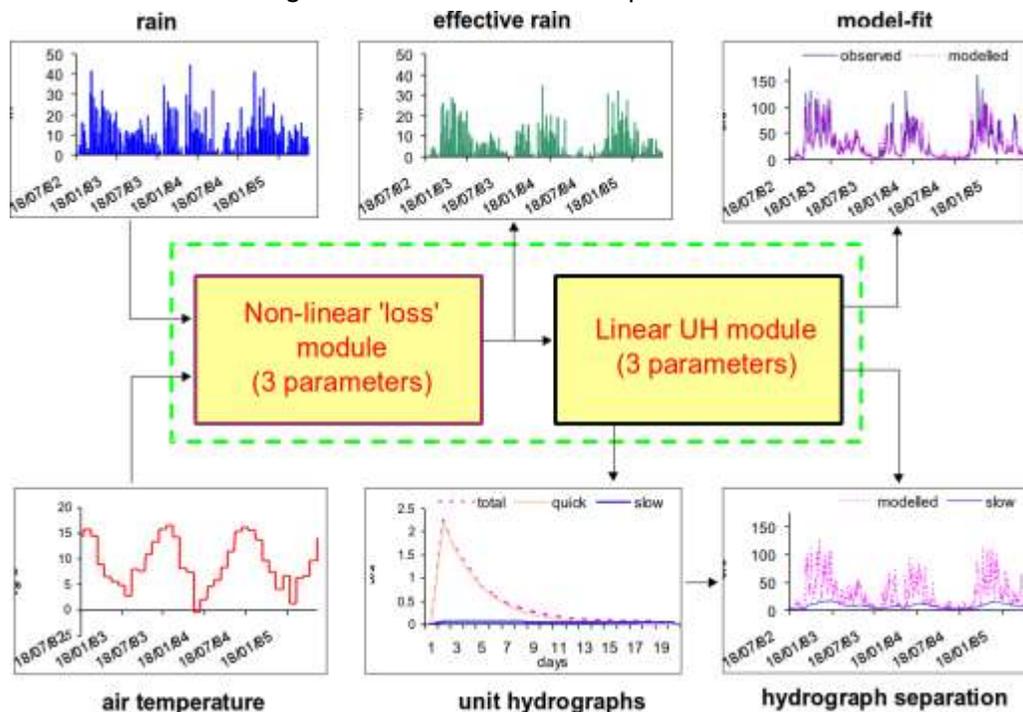


Figure 3.IX.9: Data time steps from six minutes to one month have been employed successfully by IHACRES for catchments varying in size from 490 m² (in China) to nearly 10,000 km² (in the UK).

The methodology yields a unit hydrograph (UH) for total streamflow. This can often be resolved into separate unit hydrographs that characterise the quick and slow components of streamflow. This feature allows hydrographs to be separated into their dominant quick and slow flow components and provides a Slow Flow Index (SFI) analogous to the well-known Baseflow Index (BFI).

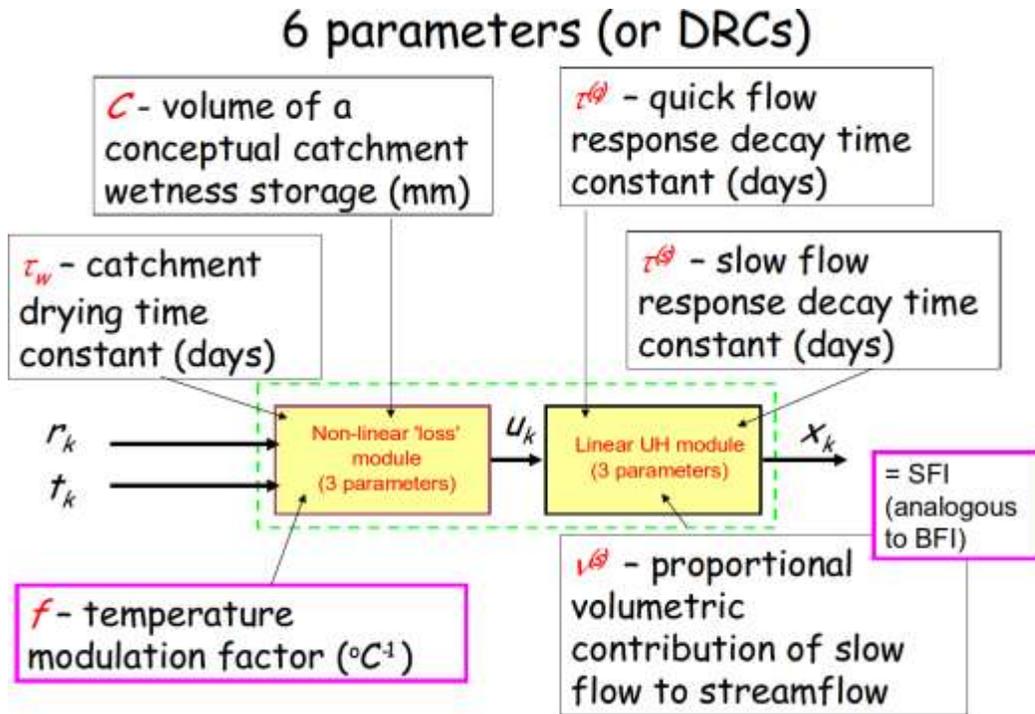


Figure 3.IX.10: Description of 6 parameters input/output of IHACRES module.

Simulation - Modelling the Human Impact on Nature -

A simple litter system: It is a model using the euler method. Carbon storage C is calculated in dependency of litter fall rate (C_{input}) and rate constant for litter loss (C_{loss}) as:

$$C_{t+1} = C_t + C_{input} - C_{loss} C_t \Delta t$$

Typical values:

Model	Parameter	Value	Unit
Tropical Rainforest	Litter fall rate	500	g/m ² /a
	Litter loss rate	2.0	1/a
Temperate forest	Litter fall rate	240	g/m ² /a
	Litter loss rate	0.4	1/a
Boreal forest	Litter fall rate	50	g/m ² /a
	Litter loss rate	0.05	1/a

Table 3.IX.8: Typical parameter for several litter system model.

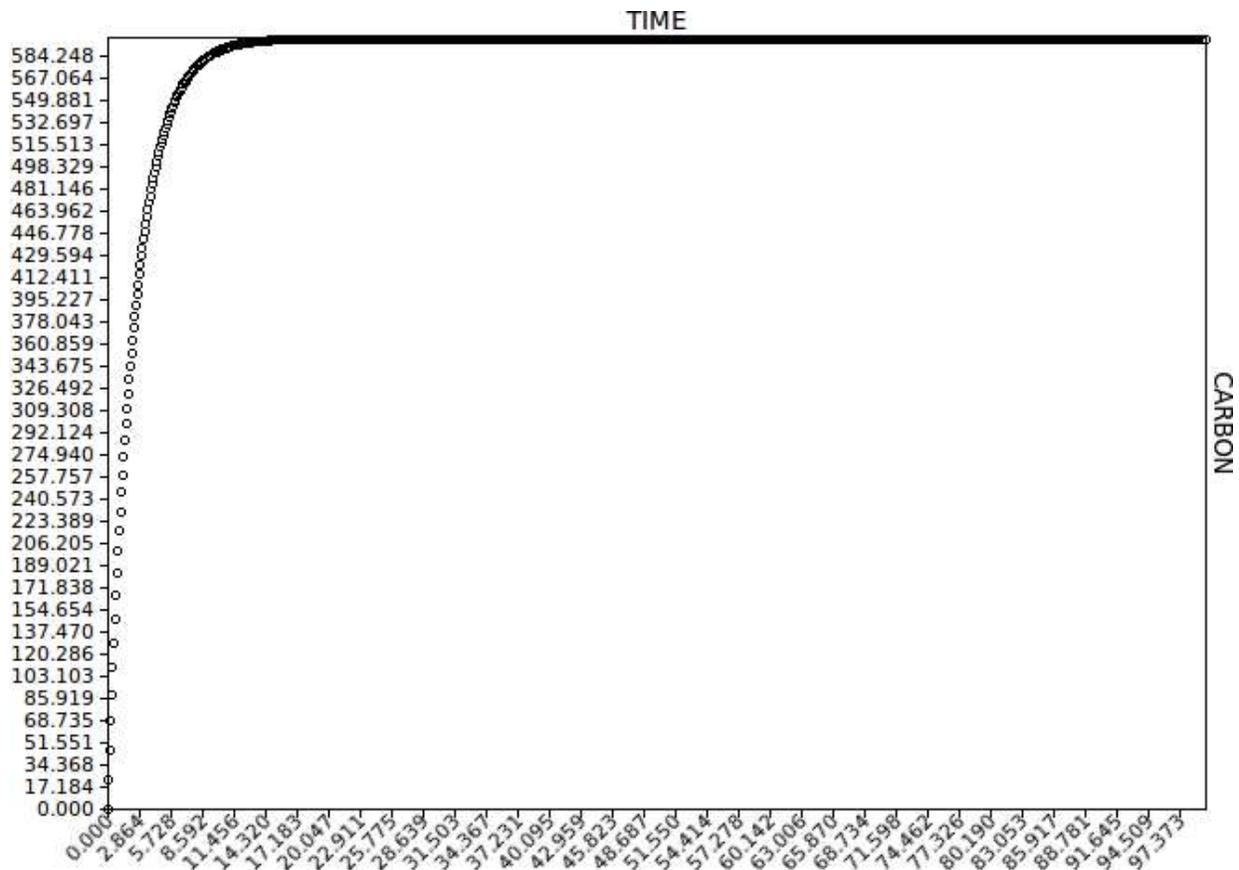


Figure 3.IX.11: Litter system simulation for temperate forest model.

Carbon Cycle Simulation for Terrestrial Biomes: Simulation of the Carbon Cycle in Terrestrial Biomes.

Spatially Distributed Simulation of Soil Nitrogen Dynamics: Spatially Distributed Simulation of Soil Nitrogen Dynamics.

Terrain Analysis - Channels -

There is a tool for digital terrain analysis. For example, taking the following DEM (figure 3.IX.12), we calculated some of the maps listed below:

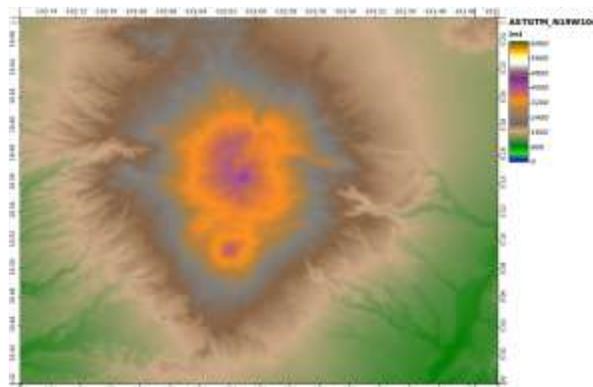
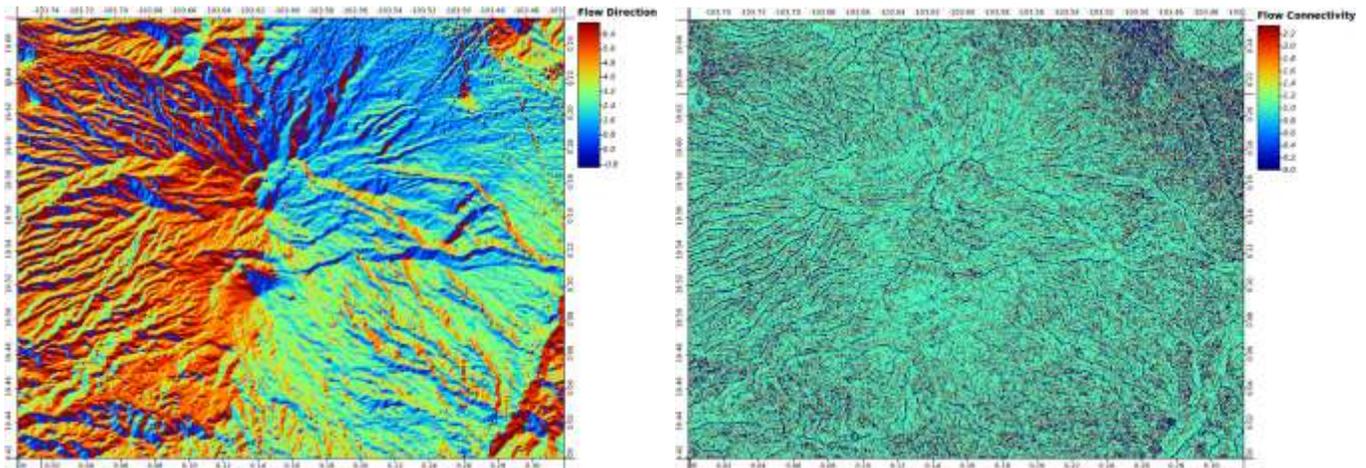


Figure 3.IX.12: DEM for channels calculations.

- **Channel Network:** This module derives a channel network based on gridded digital elevation data. Use the initiation options to determine under which conditions channels shall start.
- **D8 Flow Analysis:** Deterministic 8 based flow network analysis.



(a) (b)
Figure 3.IX.13: It shows (a) flow direction and (b) flow connectivity.

- **Overland Flow Distance to Channel Network:** This module calculates overland flow distances to a channel network based on gridded digital elevation data and channel network information. The flow algorithm may be either Deterministic 8 (O'Callaghan & Mark 1984) or Multiple Flow Direction (Freeman 1991).
- **Strahler Order:** A fast recursive algorithm for computing Strahler Stream Order. Stream ordering is a useful property of every river network, having a wide range of applications.

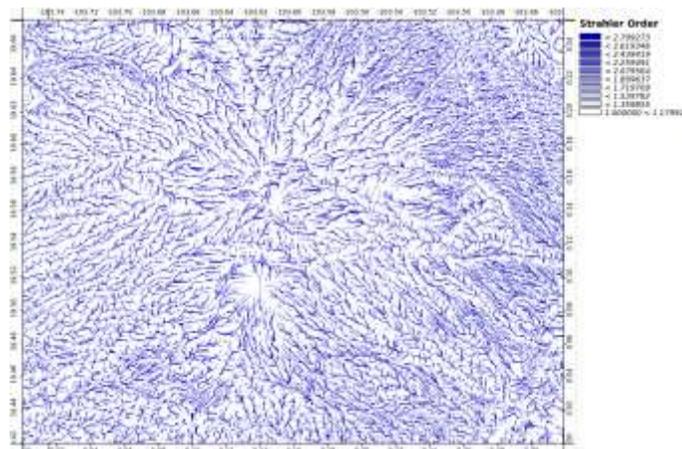


Figure 3.IX.14: Terrain Analysis, Strahler order calculation.

- **Vertical Distance to Channel Network:** This module calculates the vertical distance to a channel network base level. The algorithm consists of two major steps:
 1. Interpolation of a channel network base level elevation grid.
 2. Subtraction of this grid from the original elevations.

- Watershed Basins.
- Watershed Basins (extended).

Terrain Analysis - Hydrology -

- Cell Balance: Cell Balance Calculation.

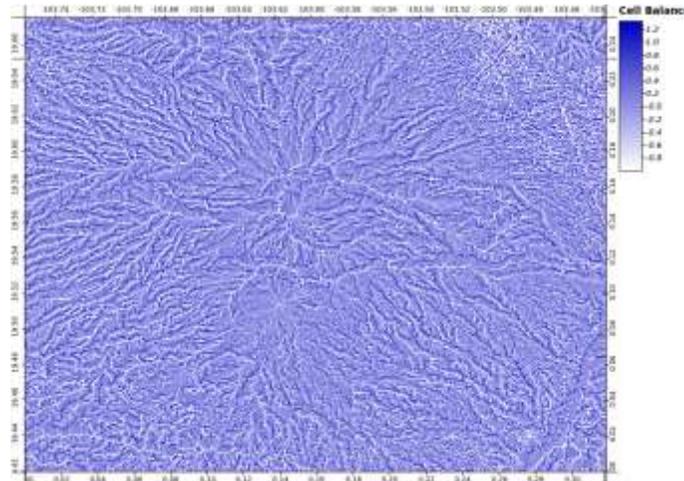


Figure 3.IX.15: It shows cell balance map.

- **Downslope Area [Interactive]:** This interactive module allows you to specify source cells (with a left mouse click), for which the downslope area shall be identified. For the 'Deterministic Infinity' and 'Multiple Flow Direction' algorithms, which are able to simulate flow divergence, the result will give the percentage of the source cell's flow that drains through each cell. Currently available algorithms are:
 - Deterministic 8
 - Rho 8
 - Braunschweiger Reliefmodell
 - Deterministic Infinity
 - Multiple Flow Direction
 - Kinematic Routing Algorithm
 - DEMON
- **Edge Contamination.**
- **Flow Depth [Interactive]:** Flow Depth Calculation.
- **Flow Path Length:** This module calculates the average flow path length starting from the seeds, that are given by the optional 'Seeds' grid and optionally from cells without upslope contributing areas (i.e. summits, ridges). Seeds will be all grid cells, that are not 'no data' values. If seeds are not given, only summits and ridges as given by the flow routing will be taken into account. Available flow routing methods are based on the 'Deterministic 8 (D8)' (Callaghan and Mark 1984) and the 'Multiple Flow Direction (FD8)' (Freeman 1991, Quinn et al. 1991) algorithms.

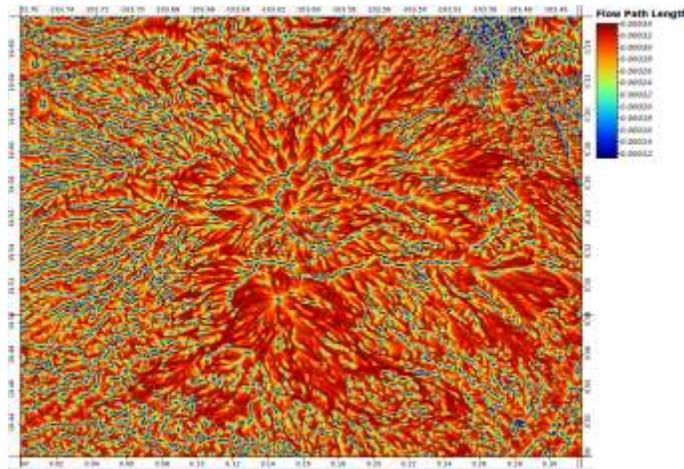


Figure 3.IX.16: Flow path length map.

- **Flow Sinuosity [Interactive]:** Flow sinuosity calculation.
- **Flow Tracing:** Flow tracing algorithms for calculations of flow accumulation and related parameters. These algorithms trace the flow of each cell in a DEM separately until it finally leaves the DEM or ends in a sink. Available flow routing methods are based on Rho8, Kinematic Routing Algorithm and DEMON.
- **Isochrones Constant Speed [Interactive]:** Isochrones calculation with constant speed
- **Isochrones Variable Speed [Interactive]:** Cálculo del tiempo de salida con velocidad variable.
- **Lake Flood:** This module can be used to simulate the extent and volume of a lake for a specified water depth in a seed cell.
- **Lake Flood [Interactive]:** This module works interactively and can be used to simulate the extent and volume of a lake for a given water depth. Execute the module and use the action tool on a cell of the digital elevation model to fill the lake starting from this location. Execute the module again to terminate module operation.
- **Parallel Processing:** Parallel processing of cells for calculation of flow accumulation and related parameters. This set of algorithms processes a DEM downwards from the highest to the lowest cell.
- **Recursive Upward Processing:** Recursive upward processing of cells for calculation of flow accumulation and related parameters. This set of algorithms processes recursively all upwards connected cells until each cell of the DEM has been processed. Available Recursive Upward Processing methods are based on Deterministic 8, Rho 8, Deterministic Infinity, Multiple Flow Direction.
- **Raphael GCS Wetness Index:** The '*Raphael GCS Wetness Index*' is, as the name says, similar to the 'Topographic Wetness Index' (TWI), but it is based on a modified catchment area calculation ('Modified Catchment Area'), which does not think of the flow as very thin film. As result it predicts for cells situated in valley floors with a small vertical distance to a channel a more realistic, higher potential soil moisture compared to the standard TWI calculation.
- **Slope Length.**
- **Topographic Indices:** Calculation of slope and catchment area based topographic indices Topographic Wetness Index (TWI), USLE LS factor, Stream Power Index.
- **Upslope Area:** This module allows you to specify target cells, for which the upslope contributing area shall be identified. The result will give for each cell the percentage of its flow that reaches the target cell(s). Available Upslope Area methods are based on Deterministic 8, Deterministic Infinitum, Multiple Flow, Direction.

- **Upslope Area [Interactive]:** This module allows you to specify target cells, for which the upslope contributing area shall be identified. The result will give for each cell the percentage of its flow that reaches the target cell(s). Available Upslope Area methods are based on Deterministic 8, Deterministic Infinitum, Multiple Flow, Direction.
- **Flow Width:** Flow width and specific catchment area (SCA) calculation. Available flow width methods are based on Deterministic 8, Multiple Flow Direction and Aspect.

Terrain Analysis - Lighting, Visibility -

Lighting and visibility calculations for digital terrain models.

- **Analysis Hillshading:** Analytical hillshading calculation. It has as input data to file elevation (DEM) and it yields a grid with angle value between the surface and the incoming light beams, measured in radians. Available methods such as: Standard, Standard (max 90 degree), Combined shading, ray tracing.
- **Incoming Solar Radiation:** Calculation of the potential incoming solar radiation. The model calculates the total solar radiation (kWh/m^2) and the duration of insolation (hours) for a given range of days.

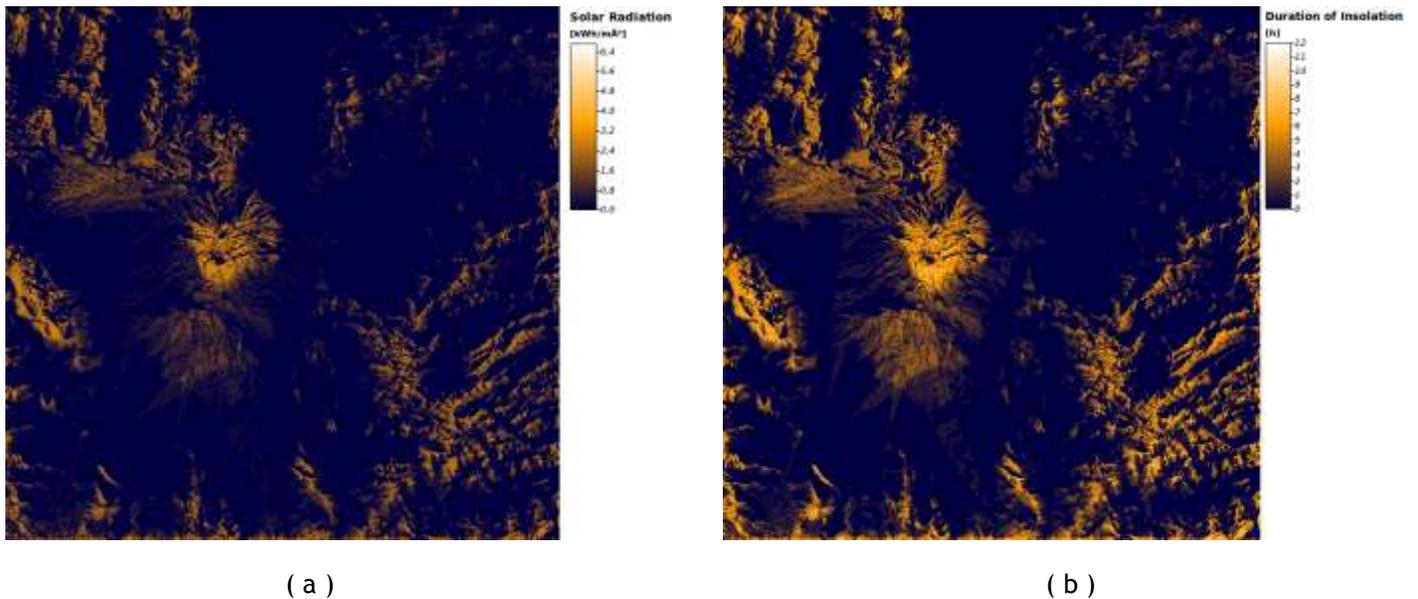
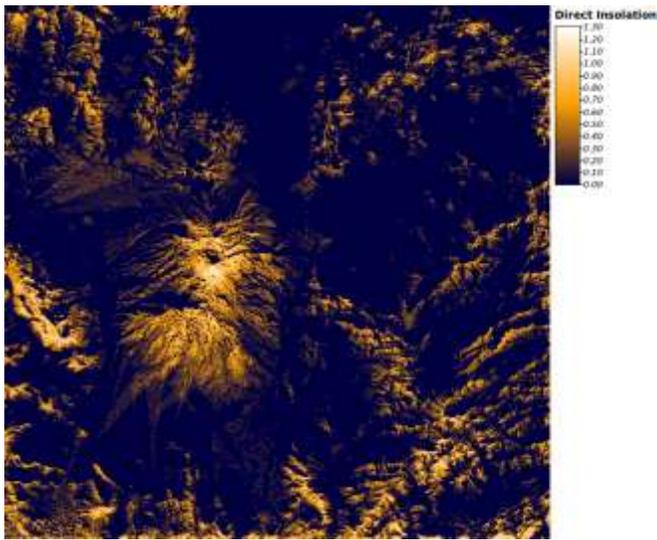
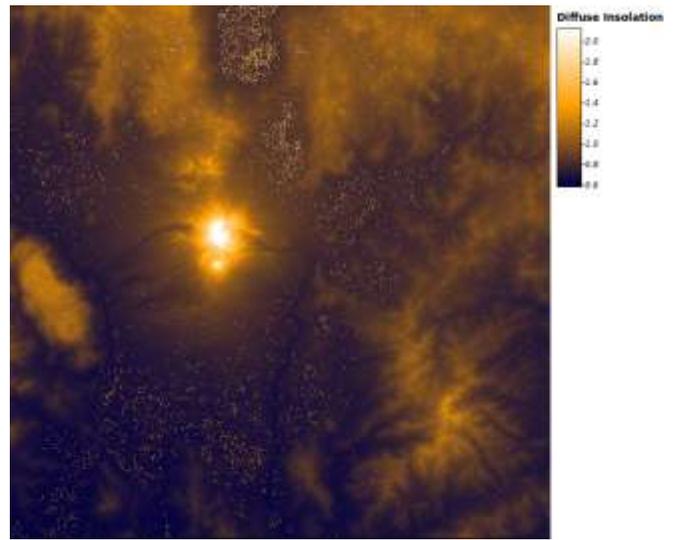


Figure 3.IX.18: Incoming solar radiation. (a) The solar radiation distribution spread on the DEM. (b) Duration of insolation spread on the DEM.

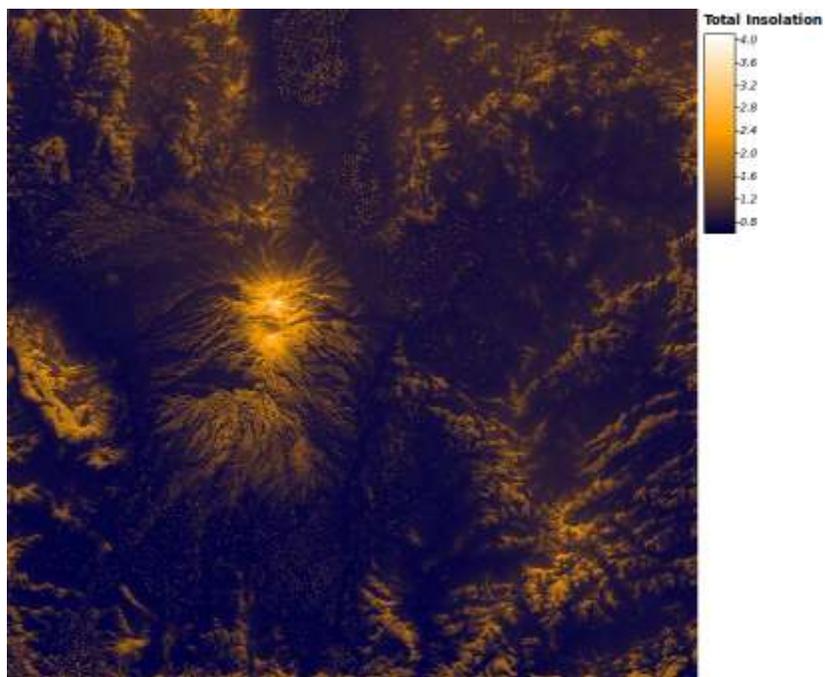
- **Insolation:** Calculation of incoming solar radiation (insolation). Based on the SADO (System for the Analysis of Discrete Surfaces) routines developed by Boehner & Trachinow. It calculates the direct insolation, diffuser insolation and the total insolation.



(a)



(b)



(c)

Figure 3.IX.20: Insolation given the de SADO method. (a) Direct insolation spread over the terrain. (b) Diffuse component spread over the terrain. (c) Total contribution of solar radiation spread over the terrain.

- **Sky View Factor:** This module calculates the Visible Sky and Sky View factor for a given DEM file.
- **Visibility (single point) [Interactive].**
- **Topographic Correction:** Topographic correction for differential illumination effects. Available methods such as: 1) Cosine Correction (Teillet et al. 1982), 2) Cosine Correction (Civco 1989), 3) Minnaert Correction, 4) Minnaert Correction with Slope (Riano et al. 2003), 5) Minnaert Correction with Slope (Law and Nichol 2004), 6) C Correction.

Terrain Analysis - Morphometry -

- **Convergence Index:** This module calculates the convergence index of a given DEM file. Available methods such as: Aspect or Gradient.
- **Convergence Index (rearch radius):** This module calculates the convergence index of a given DEM file. Available methods such as: 1) Standard, 2) Distance Weighted (Linear) and 3) Distance Weighted (Inverse).
- **Curvature Classification:** Surface curvature based terrain classification.

0	V/V
1	GE/V
2	X/V
3	V/GR
4	GE/GR
5	X/GR
6	V/X
7	GE/X
8	X/X

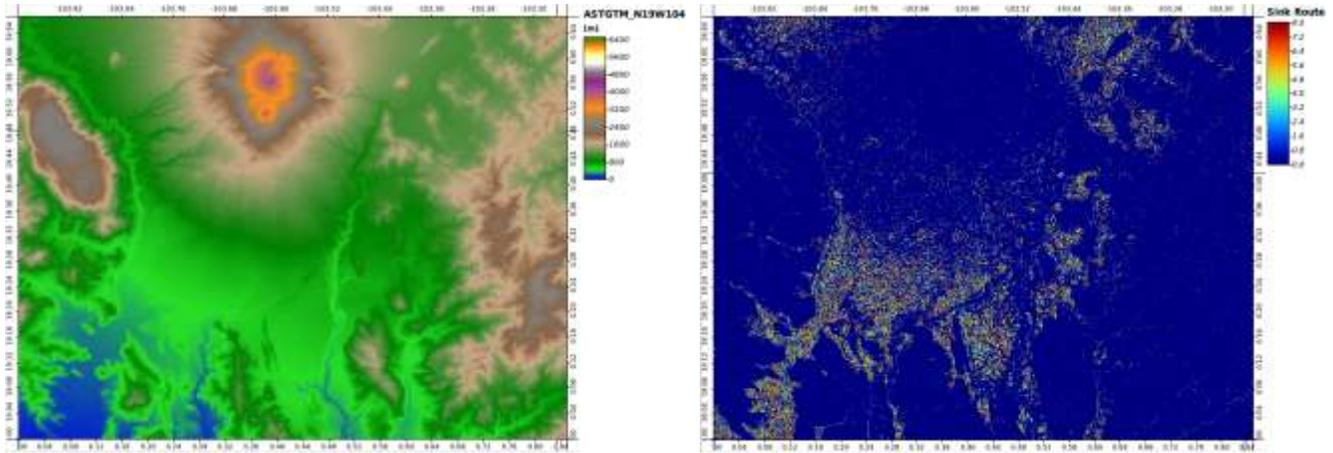
Table 3.IX.9: Curvature classification.

- **Diurnal Anisotropic Heating.** This module calculates the diurnal anisotropic heating of a given DEM file.
- **Downslope Distance Gradient:** Calculation of a new topographic index to quantify downslope controls on local drainage.
- **Effective Air Flow Heights:** This module calculates the effective air flow heights of a given DEM file. It is suppose a wind direction, Lee factor, Luv factor and maximun distance.
- **Hypsometry:** Calculates the hypsometric curve for a given DEM.
- **Land Surface Temperature:** This module calculates the land surface temperature of a given DEM, but it supposes short ware radiation grid (kW/m²) and leaf area index grid.
- **Local Morphometry:** Calculates local morphometric terrain attributes (i.e. slope, aspect and curvatures). Available methods such as: Available Choices: 1) Maximum Slope (Travis et al. 1975), 2) Maximum Triangle Slope (Tarboton 1997), 3) Least Squares Fitted Plane (Horn 1981, Costa-Cabral & Burgess 1996), 4) Fit 2.Degree Polynom (Bauer, Rohdenburg, Bork 1985), 5) Fit 2.Degree Polynom (Heerdegen & Beran 1982), 6) Fit 2.Degree Polynom (Zevenbergen & Thorne 1987), 7) Fit 3.Degree Polynom (Haralick 1983).
- **Mass Balance Index:** This module calculates the mas balance index of a given DEM and optionally as input data the vectical disntace to channel network is taken into account.
- **Morphometric Protection Index:** This module calculates the protection index of a given DEM.

- **Multiresolution Index of Valley Bottom Flatness (MRVBF):** Calculation of the 'multiresolution index of valley bottom flatness' (MRVBF) and the complementary 'multiresolution index of the ridge top flatness' (MRRTF) of a given DEM.
- **Real Area Calculation:** Calculates real (not projected) cell area.
- **Relative Heights and Slope Positions.**
- **Surface Specific Points:** Detection of surface-specific points by local parallel processing of discrete terrain elevation data. **Available methods such as:** 1) Mark Highest Neighbour, 2) Opposite Neighbours, 3) Flow Direction, 4) Flow Direction (up and down) and 5) Peucker and Douglas.
- **Wind Effect (Windward/Leeward Index):** This module calculates the wind effect of a given DEM. The input parameters are wind direction and maximum distance.

Terrain Analysis - Preprocessing

- **Fill Sinks (Planchon/Darboux, 2001):** Depression filling algorithm given Olivier Planchon & Frederic Darboux (2001). This algorithm consists of increasing the elevation of pixels in closed depressions until the sink disappears and a minimum slope angle of minslope (default: 0.01 degree) is established.
- **Fill Sinks (Wang & Liu):** This module uses an algorithm proposed by Wang & Liu to identify and fill surface depressions in digital elevation models. The method was enhanced to allow the creation of hydrologic sound elevation models, i.e. not only to fill the depression(s) but also to preserve a downward slope along the flow path. If desired, this is accomplished by preserving a minimum slope gradient (and thus elevation difference) between cells. This is the fully featured version of the module creating a depression less DEM, a flow path grid and a grid with watershed basins. If you encounter problems processing large data sets (e.g. LIDAR data) with this module try the basic version (Fill Sinks XXL).
- **Fill Sinks XXL (Wang & Liu):** This module uses an algorithm proposed by Wang & Liu to identify and fill surface depressions in digital elevation models. The method was enhanced to allow the creation of hydrologic sound elevation models, i.e. not only to fill the depression(s) but also to preserve a downward slope along the flow path. If desired, this is accomplished by preserving a minimum slope gradient (and thus elevation difference) between cells. This version of the module is designed to work on large data sets (e.g. LIDAR data), with smaller datasets you might like to check out the fully featured standard version of the module.
- **Sink Drainage Route Detection:** This module implements the sink drainage route detection algorithm. For non-sinks obtain a value of 0, sinks are assigned an integer between 0 and 8 indicating the direction to which flow from this sink should be routed, (0 = north, 1 = northeast, 2 = east, ..., 7 = northwest).



(a) (b)
 Figure 3.IX.21: Sink Drainage Route Detection, (a) DEM image, (b) Sink Route.

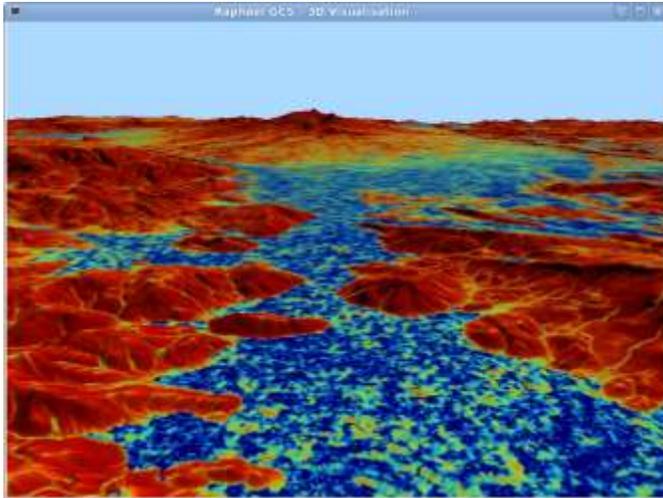
- **Sink Removal:** Remove sinks from a digital elevation model by deepening drainage routes or filling sinks.

Terrain Analysis - Profiles

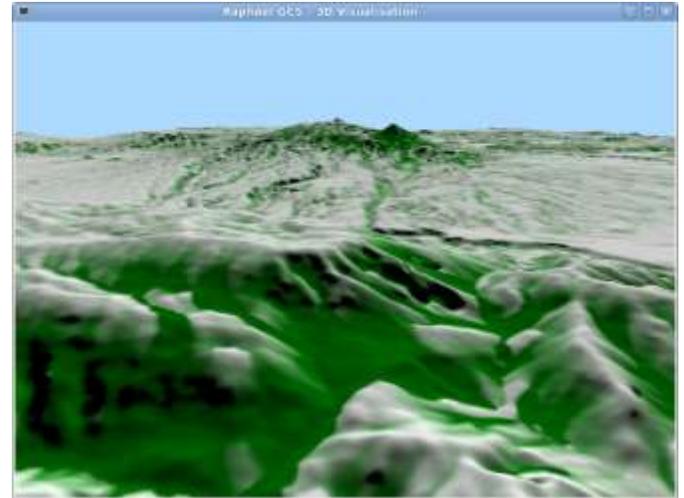
- **Cross Profiles:** Create cross profiles from a grid based DEM for given lines.
- **Cross Sections:** Creates cross section from a grid based DEM for given lines.
- **Flow Path Profile [Interactive]:** Create interactively flow path profiles from a grid based DEM. Use a left mouse button click to create a flow profile starting from the clicked point.
- **Profile [Interactive]:** Create interactively profiles from a grid based DEM. Use left mouse button clicks into a map window to add profile points. A right mouse button click will finish the profile.
- **Profile from Points:** Creates a profile from point coordinates stored in a table.
- **Profiles from Lines:** Create profiles from a grid based DEM for each line of a lines layer.
- **Swath Profile [Interactive]:** Create interactively swath profiles from a grid based DEM. Use left mouse button clicks into a map window to add profile points. A right mouse button click will finish the profile. Generated outputs for the swath profile are arithmetic mean, minimum, maximum values and the standard deviation.

3D Visualisation

All the previous images and indeed any DEM data can also be displayed in 3D as shown below for a few previously displayed cases:



(a)



(b)

Figure 3.IX.22: It shows catchment & height

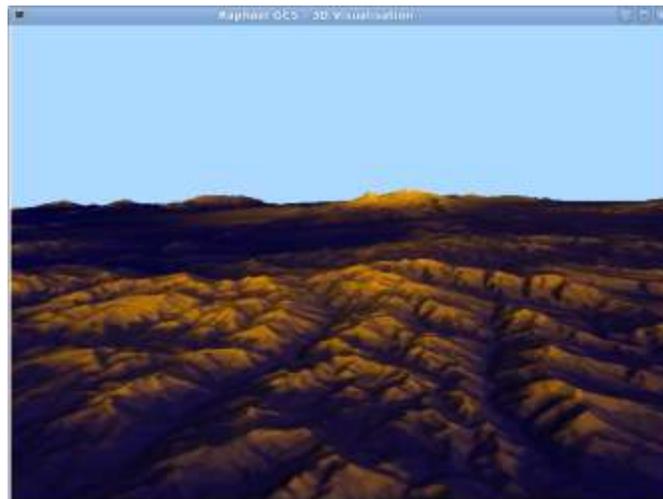


Figure 3.IX.23: Total contribution of solar radiation spread over the terrain.

4. Optional Modules

4.1 3D RECONSTRUCTION FROM FULL MOVING VIDEO (FMV)

This is an automatic algorithm capable of extracting a 3D target surface model from FMV. It automatically calculates relative camera position at each frame and tracks multiple features in order to generate a points-cloud, from which a 3D surface is formed. Multiple video feeds can be used to carry out the reconstruction of 3D target data. Occlusion/Obscuration is also contemplated and be accommodated in cases of partial obscuration, as these lead to a lack of correspondence of features being tracked.

Although we have produced a number of robotics systems for such companies as Hitachi Global Storage Technologies, in our view stereo vision should only be employed where distance-to-object is short and relative speeds high, as in in-door robotics. In cases where there is large relative movement and distance between the imaging system and the object of interest we have found that an automatic 3D reconstruction algorithm is better than a stereoscopic approach, as the distance between any two views can be varied, whereas in a stereo system this is fixed, which causes large errors as these are inversely related to the angle subtended between the two cameras and the object.

We have developed an advanced system for *3D scene reconstruction from multiple image views*, such as those obtain from full-moving videos (FMV) onboard unmanned vehicles (UAVs). If a scene of interest is seen from several viewing cameras (with different calibration parameters and physical characteristics) positioned around the target, even distinct illumination degrees, occlusion, scale, image resolution, share effect or motion, we have the only commercially available computational package capable of getting a high-definition 3D scene reconstruction. This 3D scene reconstruction work for multiple image views of a target of interest, where the position of each viewing image is only roughly known, is ideal when digital images (including photographs) are available, for example because the phenomenon is unique or of difficult access such as an asteroid, perhaps videos, or a combination of aerial videos taken at different times from varying positions with a range of imaging systems and/or ground images of a target of interest. When a sophisticated/expensive/heavy sensor such as a Light Detection and Ranging technology (LIDAR) cannot be readily accessed due to cost, availability or maximum payload constraints of a platform such as a tactical UAV, even then we can get an accurate 3D scene reconstruction. This approach will be widely useful in coming years.

Our software automatically computes information about location and orientation of the cameras only from the images themselves using computer vision methods, though providing rough GPS information certainly speeds up the process. The algorithm developed works by breaking down the problem into several stages. These are as follows: camera model and calibration is presented. Detected features points and the algorithm about matching point features between pairs of images. An iterative Structure from Motion (SfM) procedure to recover the camera parameters is used. Two methods about stereo-matching of calibrated images are also used. The Poisson reconstruction with oriented 3D reconstructed points set is employed. The final texture 3D scene is also novel. Finally, we have developed an optional and complementary module that is useful to align multiple mission tracks under distinct weather condition and illumination conditions in order to get a more detailed 3D scene reconstruction model.

This 3D scene reconstruction work for multiple image views of a target of interest, where the position of each viewing image is only roughly known, is ideal when digital images (including photographs) are available, for example because the phenomenon is unique or of difficult access such as an asteroid, perhaps videos from a youtube website, or a combination of aerial videos taken at different times from

varying positions with a range of imaging systems and/or ground images of a target of interest. When a sophisticated/expensive/heavy sensor such as a Light Detection And Ranging technology (LIDAR) cannot be readily accessed due to cost, availability or maximum payload constraints of a platform such as a tactical UAV, even thus, we can get an accurate 3D scene reconstruction using images. This approach will be widely useful in coming years.

Initially, our 3D scene reconstruction algorithm assumes multiple image views in the form of several digitally-captured 2D images of a scene or target taken from different angles and under distinct weather conditions. These images can be coded in .bmp, .png, .jpg, .gif, .mpeg, .avi, or any other digital format. Our software automatically computes information about location and orientation of the cameras only from the images themselves using computer vision methods, though providing rough GPS information certainly speeds up the process. The algorithm developed works by breaking down the problem into several stages. These will be covered as follows: in section 4.1.2 camera model and calibration is presented. Detected features points are described on section 4.1.3, and section 4.1.4 presents the algorithm about matching point features between pairs of images. Section 4.1.5 refers to an iterative Structure from Motion (SfM) procedure to recover the camera parameters. Section 4.1.6 presents two methods about stereo-matching of calibrated images. The Poisson reconstruction with oriented 3D reconstructed points set is presented in section 4.1.7. The final texture 3D scene is described in Section 4.1.8. Finally, section 4.1.9 shows an optional and complementary module that is useful to align multiple mission tracks under distinct weather condition and illumination conditions in order to get a more detailed 3D scene reconstruction model. Figure 4.1.1 shows images taken by the synthetic vision module of our ground control station *Raphael GCS* (Gutiérrez-Resendiz and Funes-Gallanzi, 2009).

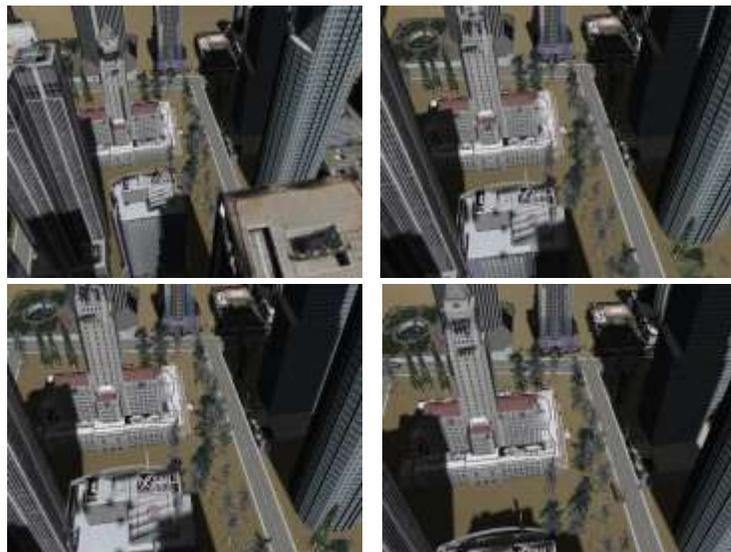


Figure 4.1.1: Four images taken from the synthetic vision toolkit of *Raphael GCS* are shown.

4.1.2.- Camera Model and calibration

As a first step, we consider a camera as an electronic device that records an image digitally. The pinhole camera model is widely used as the real camera projection. This model is based on the principle of collinearity, where each point in the object space is projected by a straight line through the projection center into the image plane. It is a useful model that enables simple mathematical

formulation for the relationship between object and image coordinates. However, it is not valid when high accuracy is required and therefore, a more comprehensive camera model must be used. The model contains parameters divided into extrinsic and intrinsic parameters. Extrinsic parameters refer to camera pose and intrinsic parameters refer to image plane properties.

The origin of the camera coordinate system is in the projection center at the location (x_0, y_0, z_0) with respect to the object coordinate system, and the z-axis of the camera frame is perpendicular to the image plane. The rotation is represented using Euler angles ω , φ , and κ that define a sequence of three elementary rotations around x, y, z-axis respectively. Rotations are performed clockwise, first around the x-axis, then the y-axis which is already once rotated, and finally around the z-axis that is twice rotated during the previous stages.

In order to express an arbitrary object point P at location $[X_i, Y_i, Z_i]$ in image coordinates, we first need to transform it to camera coordinates $[x, y, z]$. This transformation consists of a translation and a rotation, and it can be performed by using the following matrix equation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

where

$$\begin{aligned} r_{12} &= \sin \omega \sin \varphi \cos \kappa - \cos \omega \sin \kappa & r_{11} &= \cos \varphi \cos \kappa \\ r_{22} &= \sin \omega \sin \varphi \sin \kappa + \cos \omega \cos \kappa & r_{21} &= \cos \varphi \sin \kappa \\ r_{13} &= \cos \omega \sin \varphi \cos \kappa + \sin \omega \sin \kappa & r_{31} &= -\sin \varphi \\ r_{23} &= \cos \omega \sin \varphi \sin \kappa - \sin \omega \cos \kappa & r_{32} &= \sin \omega \cos \varphi \\ r_{33} &= \cos \omega \cos \varphi \end{aligned}$$

The intrinsic camera parameters usually include the effective focal length f , scale factor s_u , and the image center $[u_0, v_0]$, also called the principal point. Here, as usual in computer vision literature, the origin of the image coordinate system is in the upper left corner of the image array. The unit of the image coordinates is pixel, and therefore coefficients D_u and D_v are needed to change the metric units to pixels. In fact, their precise values are not necessary, because they are linearly dependent on the focal length f and the scale factor s_x . By using the pinhole model, the projection of the point (x_i, y_i, z_i) to the image plane is expressed as

$$\begin{bmatrix} \tilde{u}_i \\ \tilde{v}_i \end{bmatrix} = \frac{f}{z_i} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

The corresponding image coordinates (u'_i, v'_i) in pixels are obtained from the projection $(\tilde{u}_i, \tilde{v}_i)$ by applying the following transformation:

$$\begin{bmatrix} u'_i \\ v'_i \end{bmatrix} = \begin{bmatrix} D_u s_u \tilde{u}_i \\ D_v \tilde{v}_i \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

Usually, the pinhole model is a basis that is extended with some corrections for the systematically distorted image coordinates. The most commonly used correction is for the radial lens distortion that causes the actual image point to be displaced radially in the image plane. The radial distortion can be approximated using the following expression:

$$\begin{bmatrix} \Delta u_i^{(radial)} \\ \Delta v_i^{(radial)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_i(k_1 r_i^2 + k_2 r_i^4) \\ \tilde{v}_i(k_1 r_i^2 + k_2 r_i^4) \end{bmatrix}$$

where k_1 and k_2 are coefficients for radial distortion, and $r_i = (\tilde{u}_i^2 + \tilde{v}_i^2)^{1/2}$. Typically, one or two coefficients are enough to compensate for the distortion.

Curvature centers of surface lenses of lens surface are not always strictly collinear. This introduces another common distortion type, de-centering distortion that has both a radial and tangential component.

The expression for the tangential distortion is often written in the following form:

$$\begin{bmatrix} \Delta u_i^{(tangential)} \\ \Delta v_i^{(tangential)} \end{bmatrix} = \begin{bmatrix} 2p_1 \tilde{u}_i \tilde{v}_i + p_2 (r_i^2 + 2\tilde{u}_i^2) \\ p_1 (r_i^2 + 2\tilde{v}_i^2) + 2p_2 \tilde{u}_i \tilde{v}_i \end{bmatrix}$$

where p_1 and p_2 are coefficients for tangential distortion.

A sufficiently accurate camera model for accurate calibration can be derived by combining the pinhole model with the correction for the radial and tangential distortion components:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} D_u s_u (\tilde{u}_i + \Delta u_i^{(radial)} + \Delta u_i^{(tangential)}) \\ D_v (\tilde{u}_i + \Delta v_i^{(radial)} + \Delta v_i^{(tangential)}) \end{bmatrix} + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix}$$

In this model the set of intrinsic parameters $[f, s_u, u_0, v_0]$ is augmented with the distortion coefficients k_1, k_2, p_1 and p_2 . These parameters are also known as physical camera parameters, since they have a certain physical meaning. Generally, the objective of the explicit camera calibration procedure is to determine optimal values for these parameters based on image observations of a known 3D target. In the case of self-calibration, the 3D coordinates of the target points are also included in the set of unknown parameters.

Since we have the pinhole camera model for the captured images and we have calibrated each camera, we created an image list with information about image sizes and camera focal length. Since images had been captured under different conditions; this last data could not always exist, however our system may calculate an initial estimation for it, or posses a database with a family of imaging systems characterized as used in any particular theater of operations.

4.1.3.- Detection Features

In order to find feature points in each image, we use feature-matching, which is a fundamental step in 3D scene reconstruction. We have used the well-known SIFT method which is *Scale-Invariant Feature Transform* to detect and describe local features (keypoints) in images (Lowe, 2004). A typical image contains several thousand SIFT keypoints. However, sometimes the number of keypoints falls below the minimum number of keypoints to get a good estimation of initial camera pose. Then, we have adapted an interface editor to increase or modify the number of keypoints, the user can do this by hand. Each keypoint is invariant to image scale, rotation, affine distortion, change in 3D viewpoint, in addition of noise and changes in illumination. The SIFT method ensures that a single feature can be correctly matched with high probability against a large database of feature from many images. The following are the major stages of computation used to generate the set of keypoints:

1. **Scale-space extreme detection:** The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify interest points that are invariant to scale and orientation.
2. **Keypoint location:** At each candidate location, a detailed model is fitted to determine location and scale. Keypoints are selected based on measures of their stability.
3. **Orientation assignment:** One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that have been transformed relatively to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
4. **Keypoint descriptor:** The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

In addition to the keypoints location themselves, SIFT provides a local descriptor for each keypoint in the following sense:

- Consider a 16x16 patch centered on the keypoint.
- Decompose this patch into 4x4 pixel tiles.
- For each such tile, we compute a histogram of its pixels gradient orientations with 8 bins, each covering 45 degrees.
 - Actually, we specify these gradient orientations “relative” to the keypoint’s dominant orientation.
 - We weigh the contribution to the bin by the gradient magnitude.
 - We use a circular Gaussian falloff from the keypoint center (about 8 pixels, half the descriptor window).
- This gives 16 tiles x 8 histogram bins/tile = a 128-dimensional vector representing our a SIFT feature descriptor.
- Normalize this to unit length.

- For better illumination invariance a threshold value in this normalized vector to be no larger than 0.2 and then renormalize.
- This vector normalized is invariant to location, scale, rotation, shear and changes in illumination.

The figure 4.1.3.1 shows a typical output of a SIFT detection process. Each image contains thousands of SIFT features. Keypoints are shown as vectors in cyan with magnitude and direction only.

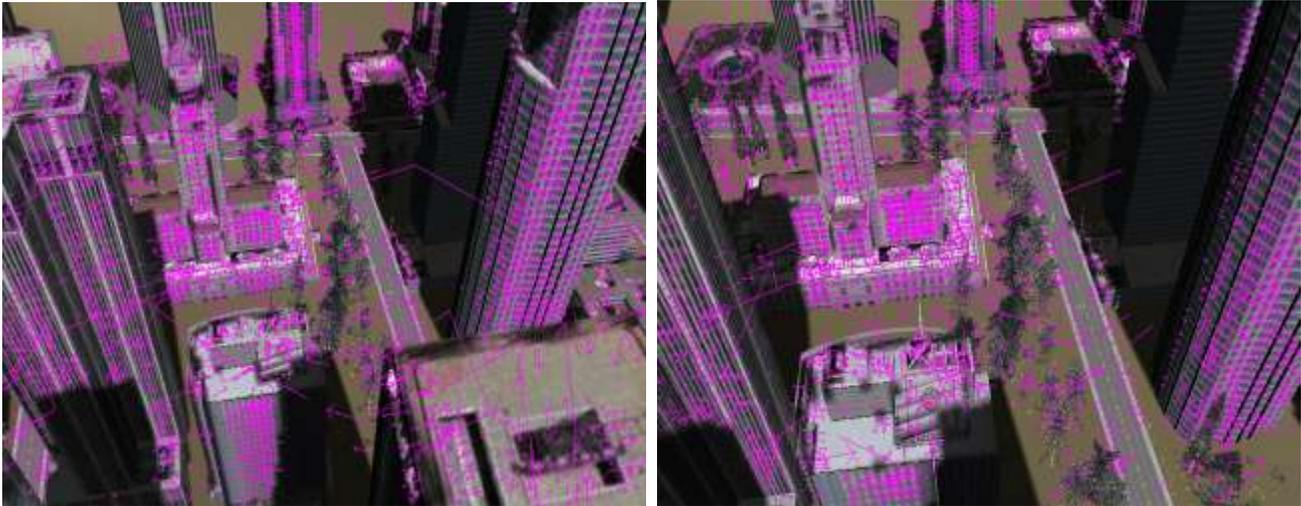


Figure 4.1.3.1: Figures show keypoints. Left) The 798 x 510 pixel image has 4467 keypoints. Right) The 798 x 510 pixel image has 3824 keypoints. The keypoints are displayed as vectors indicating scale, orientation and locations.

4.1.4.- SIFT matching between pair images

Since each keypoint is described by a vector 128-dimensional. We have used the SIFT method to get an automatic feature matching over all image pairs, using the “Approximate Nearest Neighbors” package of Arya, et. al. (1998):

- We take Euclidian distance in the space of 128-dimensional vectors.
- We use kd-trees for efficient near neighbor search.
- We compare the distance to best matching SIFT feature in the database, and the distance to the 2nd-best matching feature. If the ratio of closest distance to 2nd closest distance greater than 0.8 then reject as a false match.

The figure 4.1.4.1 shows a typical example of SIFT matching between two arbitrary chosen images. A cyan line marks the correspondence between two keypoints. However, we have included epipolar geometry constraints in order to do a best matching between keypoints as describe below.

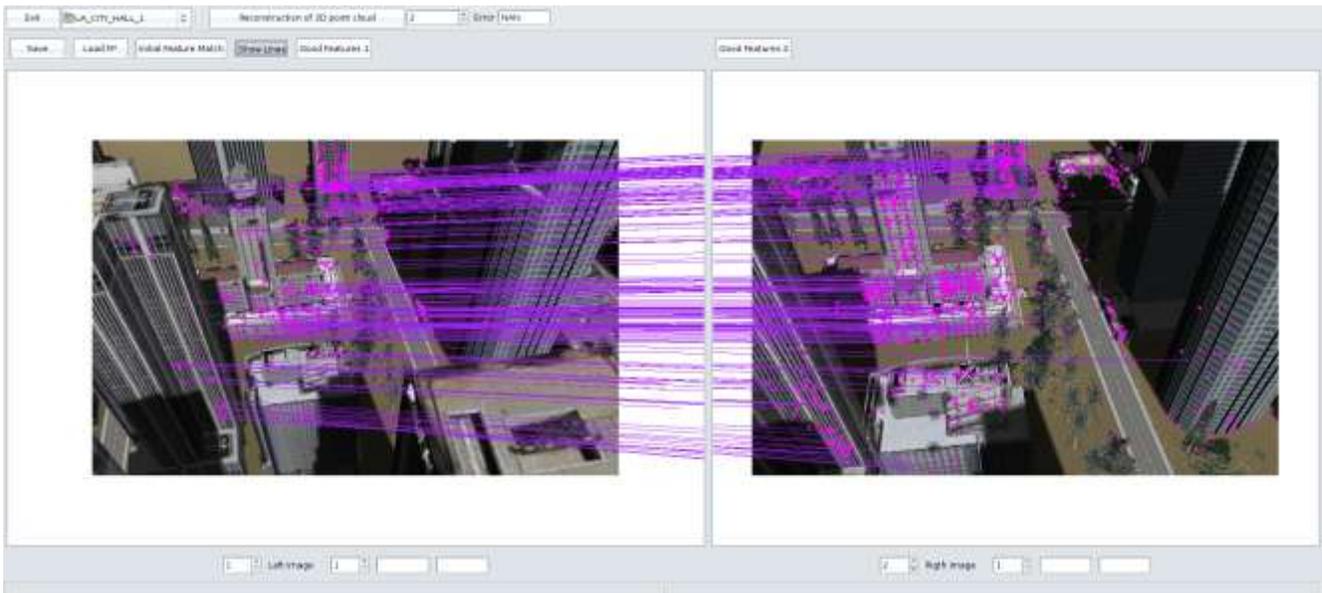


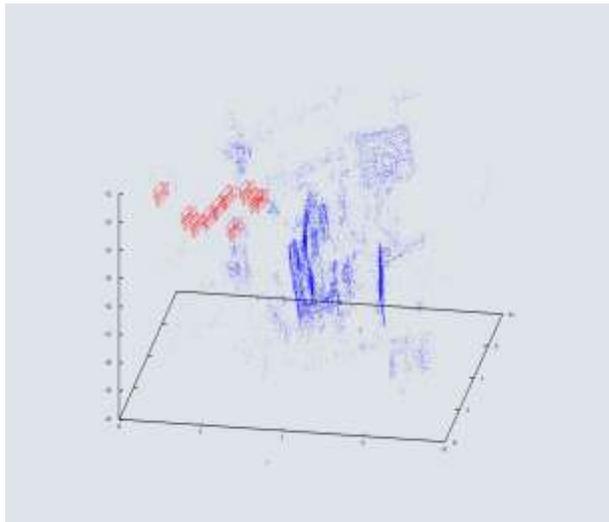
Figure 4.1.4.1: SIFT matching between image 1 and image 2. The cyan line marks the same point seen on different frames.

Once initial matching is done using a local description vector for each keypoint, then robustly we estimate the fundamental matrix F for each pair image using the RANSAC method (Fischler and Bolles 1987). As is well-known in computer vision, the fundamental matrix F is a 3×3 matrix which relates 2D corresponding points in stereo images. The fundamental matrix relates 2D coordinate images between two stereo images: $x_2^T F x_1 = 0$. Furthermore, if we know image camera parameters, we have the so-called essential matrix, $E = K_2^T F K_1$, that relates to rotation and translation between cameras: $E = R t_{[x]}$.

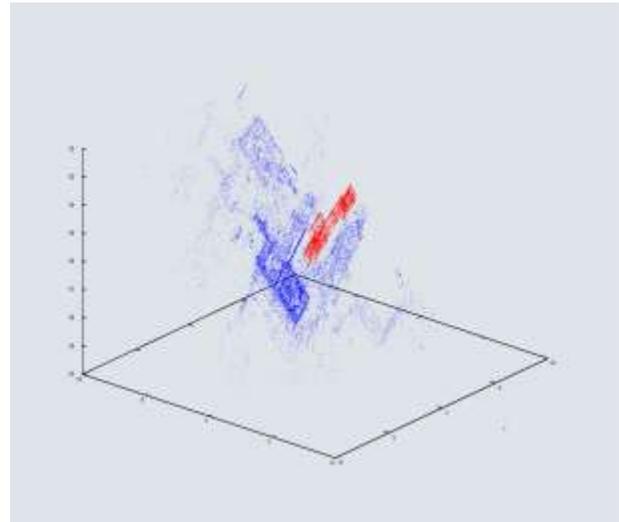
During each RANSAC iteration, we compute a candidate fundamental matrix F using the eight-point algorithm (Hartley and Zisserman 2004), followed by a non-linear refinement. The epipolar constraint is applied intensively around the SIFT matching to ensure better feature matches. We look for a set of geometrically consistent matches between each image pair in an incremental scheme to reconstruct an initial sparse 3D point-cloud and camera pose. Finally, we remove matches that are outliers to the epipolar geometry constraint. In addition, we recover an initial set of camera parameters and 3D scene reconstructed point-cloud. In order to get refinement for new camera poses and 3D scene point-cloud, we take the back-projection error as the cost function to minimize it. This is a much more computationally-intensive approach than previously reported but it is robust and able to cope with real-life images.

4.1.5.- Bundle Adjustment

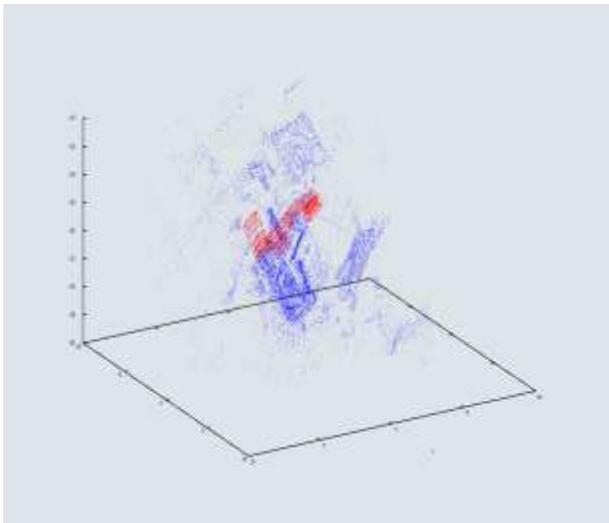
Once the last frame is added, a regularization method for final camera poses is calculated by a generic **Bundle Adjustment** method, based on the sparse Levenberg-Marquardt algorithm given by Lourakis and Argyros (2004). The Bundle Adjustment is calculated till back-projection errors convergence. The initial 2D-to-3D correspondence is refined by the Bundle Adjustment scheme, and it is used to compute the final camera pose. The 3D point-cloud scene reconstruction is carried out, using all 2D camera views and 3D camera-specific information. A typical output, is as shown in figure 4.1.5.1.



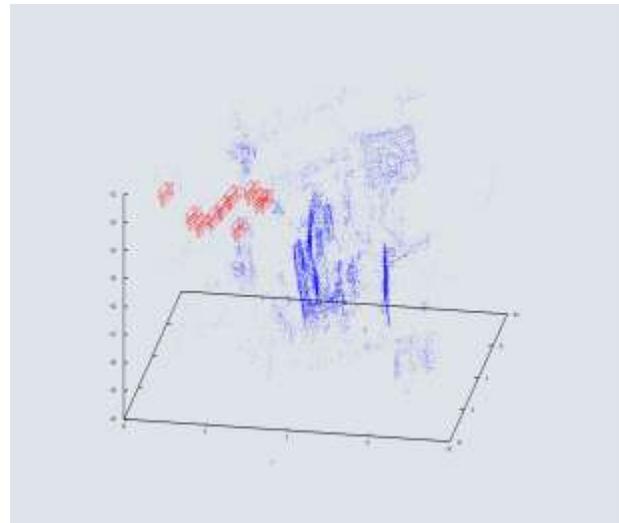
(a)



(b)



(c)



(d)

Figure 4.1.5.1: Shows output from a typical Bundle Adjustment run. Figures a), b), c) and d) show 3D point-cloud reconstruction in blue dots and camera pose in red line. We have included 10 camera poses only, for illustration purposes. See the full video on <http://www.thegoodwillcompany.co.uk>

4.1.6.- 3D Scene Surface Reconstruction

So far we have constructed a sparse 3D point-cloud of the viewed scene and to simultaneously recover camera poses, as shown in figure 6. In order to get a dense matching between multiple view images, we have combined two methods for this purpose. In order to get dense 3D oriented points, we combine two multi-view stereo matching methods: Iteratively-grown surface from matched points (Goesele et. al., 2007) and Patch-based Multi-view Stereo (Furukawa and Ponde 2007, 2009). After, the dense matching is done, we solve for the iso-surface solution in a Poisson surface reconstruction algorithm as given by Kazhdan et. al. (2006).

Multi-view stereo matching

- i) Iteratively grows surface from matched points (Goesele et. al., 2007): This stereo matching technique takes as input sparse 3D points reconstructed from bundle adjustment method and iteratively grows surfaces from these points. The main algorithm looks at both depth and normal vector, starting from an initial SIFT matched points or copied from previously computed neighbors. In order to find optimal surfaces around SIFT matches, we apply a two-level view selection camera: i) at the image level, global view selection identifies for each reference view a set of good neighborhood images to use for stereo matching. ii) at the pixel level, local view selection determines a subset of these images that yields a stable stereo match. The optimization stage for surface normals is within a photometry consistency calculation what significantly improves the matching results. We include reflectance coefficients between pair images in order to model changes in illumination. After we have run this stage, we have oriented points around SIFT matched points within a list of associated cameras, each potentially with different characteristics.
- ii) Patch-based Multi-view Stereo, PMVS (Furukawa and Ponce 2007, 2009): The patch-based multi-view stereo matching is complementary to the above stereo matching technique, i.e. we obtain additional oriented points, where the other method fails. PMVS is implemented as a match, expand and filter procedure, starting from a sparse set of SIFT matched points, and repeatedly expanding these to nearby pixel correspondences before using visibility constraints to filter away false matches:
 - 1) Matching: Features found by Harris and Difference-of-Gaussians operators are matched across multiple images, yielding a sparse set of oriented patches associated with salient image regions. Given these initial matches, the following two steps are repeated 3 times.
 - 2) Expansion: The technique is used to spread the initial matches to nearby pixels and obtain a dense set of oriented patches.
 - 3) Filtering: Visibility constraints are used to eliminate incorrect matches lying either in front or behind the observed surface.

The two methods yield a table of 3D-oriented points with a list of associated cameras. If we merged the two previous algorithms, we have a better representation of the 3D scene reconstruction. Now, we use a Poisson reconstruction algorithm in order to get the final 3D mesh associated with the final reconstructed model.

4.1.7.- Poisson Surface Reconstruction

We follow the work done by Kazhdan et. al. (2006). We assume the surface reconstruction from oriented points can be cast as a spatial Poisson problem. This Poisson formulation considers all the points at once, without resorting to heuristic spatial partitioning or blending, and is therefore highly resilient to data noise. Unlike radial basis function schemes, the Poisson approach allows a hierarchy of locally supported basis functions, and therefore the solution reduces to a well-conditioned sparse linear system. Poisson surface reconstructions generate a mesh from a set of surface samples. After Poisson reconstruction is performed for the previous stereo matching step, we get a refined surface. Figure 7 shows the surface obtained from the 3D-oriented points given by the merged stereo matches techniques above.

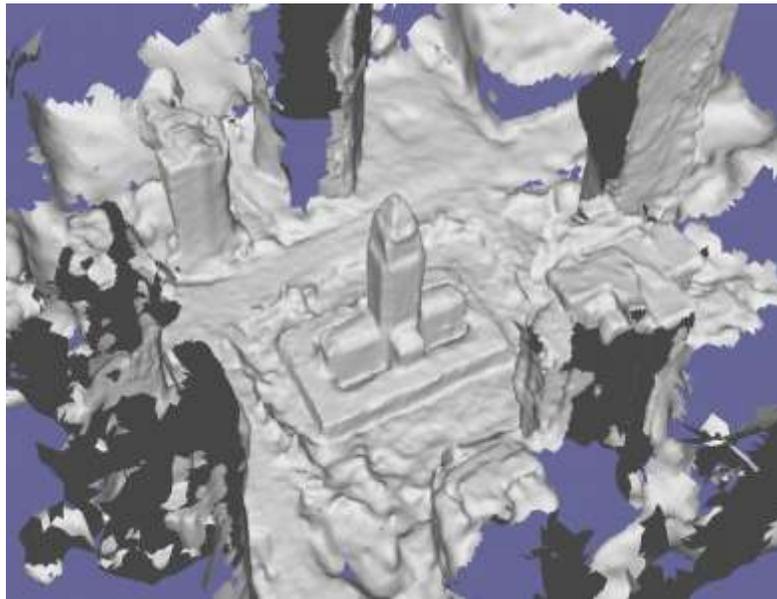


Figure 4.1.7.1. Poisson surface reconstruction of the previous stereo-matched mesh.

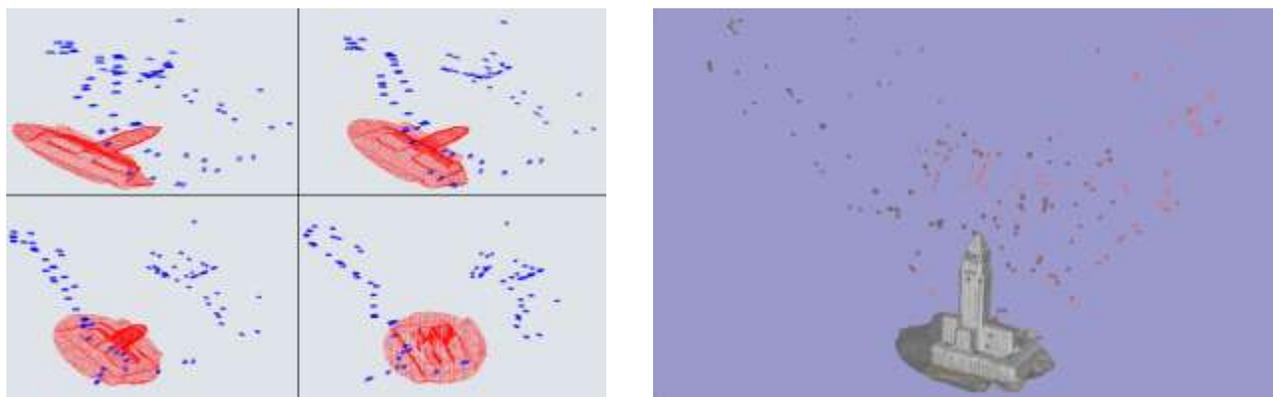
4.1.8.- Textured 3D reconstructed scene

The texture is taken from the camera views and combined to form a multi-texture surface. In order to take into account the appropriate texture of the 3D reconstructed model, we follow two steps: i) Each reconstructed 3D point has a list of associated cameras, given by the SIFT matching process and the bundle adjustment process and ii) each 3D reconstructed point has a list of visibility after the stereo matching process is done. Thus, we have at least two criteria for the visibility for a given 3D reconstructed point. Since the stereo matching process has an implicit high degree of photometric consistency, we combine visibility information to take the camera that has the maximum, color cross-correlation within the cameras list in a small oriented patch around each 3D reconstructed point. We check RGB color continuity around each orientated patch in order to assign texture on it. The figures 4.1.8.1 shows the textured 3D reconstructed model directly compared to the original CAD model. In

addition, the figure 4.1.8.2 shows the 3D scene reconstruction and it displays some of the camera poses. The figure 4.1.8.3 shows the whole 3D scene reconstruction.



(a) (b)
Figure 4.1.8.1: Original CAD building (a) compared to its 3D reconstruction from FMV (b).



(a) (b)
Figure 4.1.8.2: Four views of the FMV camera pose sequences (a) and whole-scene textured view (b).

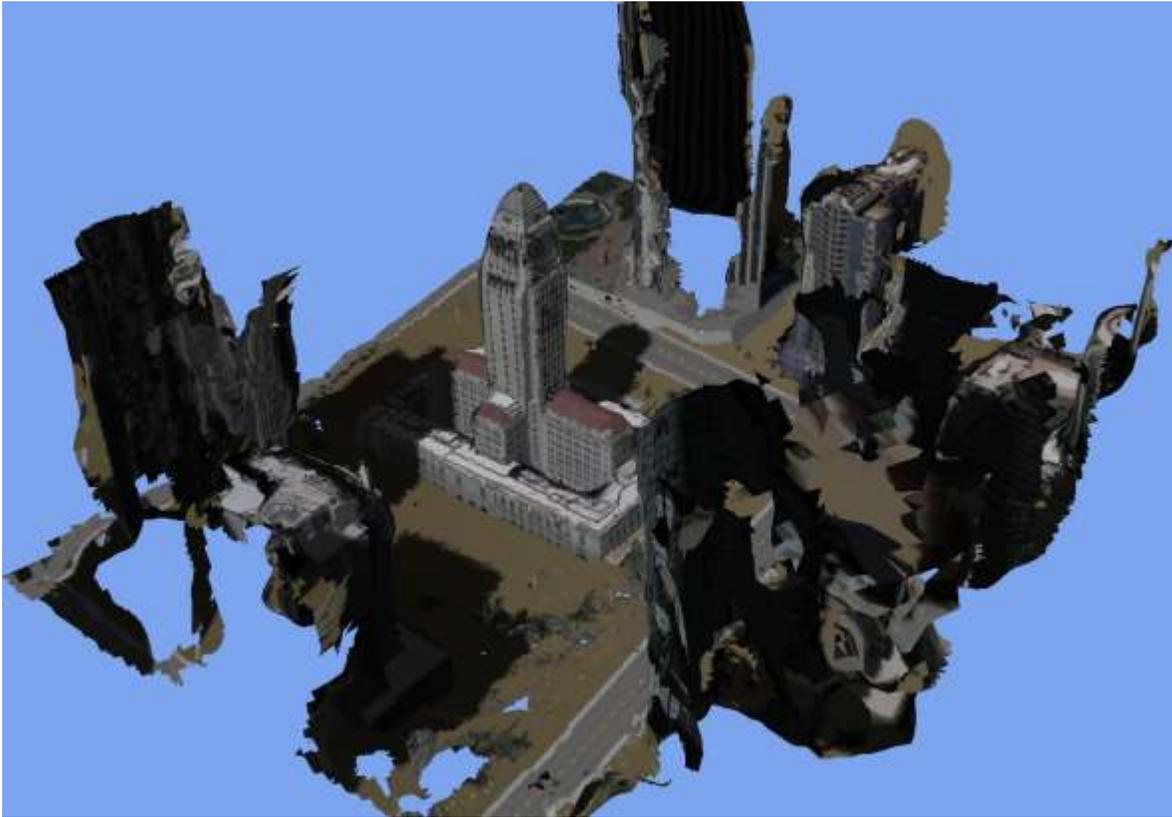


Figure 4.1.8.3: Final 3D reconstruction.

4.1.9.- Alignment between independent mission tracks

We also take into account that when we have independent mission tracks under distinct operational conditions, we have additional information to get a more detailed 3D reconstruction, since the new images can be seen from different points of view and different illumination conditions, even using distinct imaging devices. This fact allows us to write down an extra module that aligns independent 3D reconstructions given by distinct mission tracks and it gets better detailed 3D reconstruction with appropriate texture with reduced illumination effects.

In order to overlap two 3D oriented point sets, we have used a modified version of the Iterative Closest Point (ICP) algorithm developed by Besl and McKay (1992), Chen and Medioni (1992) and Zhang (1994). A classical ICP is based on the search for pairs of nearest points in two data sets and estimates the rigid body transformation that aligns them. The rigid body transformation is then applied to the points of one set and the procedure is iterated until convergence is achieved. This point-to-point scheme has low convergence. Thus, we have used a modified version of the ICP, the point-to-tangent plane approach originally proposed by Potmesil(1983).

The search for pairs of nearest points in two data sets, is done using a kd-trees structure. Thus, the Approximate Nearest Neighbors package of Arya, et. al. (1998) is used extensively in this process. On the other hand, the rigid body transformation is given by a translation vector $T(x, y, z)$ and a matrix of rotation $R(\omega, \varphi, \kappa)$. In order to find the correct rigid body transformation, we have used a Levenberg-Marquardt algorithm as the least square method to solve for the translation vector and

rotation matrix. Furthermore, we have included in the least square problem information about normal vectors over each matched pair point.

Finally, we get an enlarged list of 3D orientated points with additional information about mixed cameras. This new list of 3D orientated points is filtered out using the Poisson reconstruction process to get a new 3D surface reconstructed model with additional detail given by the combination of multiple and independent mission tracks. Since we have more 3D orientated points with associated camera-lists, we can get a better texture 3D surface model.

Of course, this extra images could include images taken from the ground, thus being complementary to those taken from satellites, LIDAR and unmanned vehicles in more than one mission and at different times in order to derive a high-quality 3D reconstruction of any target of interest.

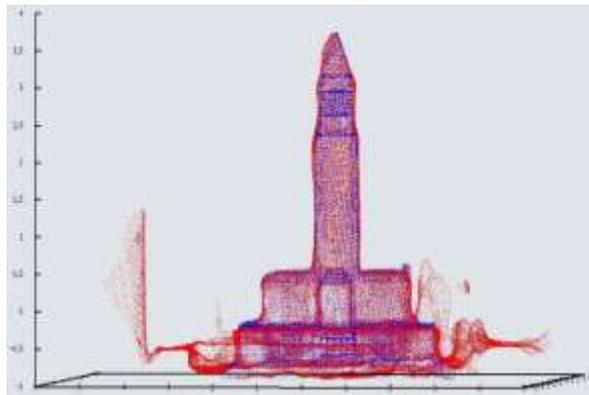


Figure 4.1.9.1: The alignment between a oriented 3D reconstructed mesh (red dots) and the CAD model mesh (blue dots).

Comparison between CAD model and 3D reconstructed scene

We have tested this algorithm on a number of real-life objects but here we show a study of a building of known dimensions in order to quantitatively estimate the algorithm's accuracy. We have quantified our 3D reconstruction process through a typical 3D scene reconstruction with a well-known CAD model. This is done within the synthetic vision module of *Raphael GCS* in order to compare the 3D surface reconstruction and the original 3D mesh model point set. We have taken the template and fixed and fitted the 3D reconstructed mesh using a modified version of the ICP algorithm as above. We search in an iterative process for a rigid body transformation that includes normal vector information between this two point sets, and we get the final overlapping mesh between template and 3D reconstructed model.

In figure 4.1.9.1 we show the overlap between the 3D reconstructed model (red dots) and the its well-known CAD model (blue dots). We read in the CAD model from an .OBJ file. As we can see the agreement is excellent. Thus, we calculated a histogram of error associated over each axis. Figure 11 shows a histogram of the ratio between the shortest distance between both model and 3D reconstruction divide by a characteristic scale, in this case the building height. The average standard deviation about the three axes is around 1.0%.

CAD Data vs. Multiple Images

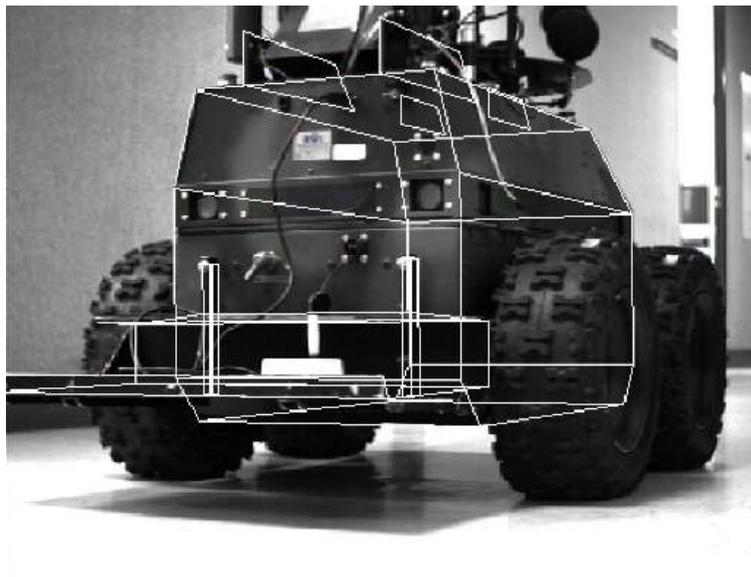
In many applications, such as aircraft ID, once a 3D object has been reconstructed, it needs to be positively identified, typically against a library of possible candidates. This module is able to compare objects embedded within multiple images with a library of CAD objects to enable the determination of a specific item and item orientation. The *Raphael GCS* module is based on the SoftPOSIT algorithm which is a fast iterative method for estimate the pose of a 3D object using multiple images even in the case that correspondences between model features and image features are unknown.



(a)



(b)



(c)

Figure 4.1.9.2: (a) Shows a single image, (b) Shows initial pose of the CAD model over the image and (c) shows the estimate pose of the CAD model versus a single image.

4.II MULTISPECTRAL ANALYSIS

This module mainly works with multispectral satellite metadata. Satellite information is normally written using hdf format version 4 or 5. The hdf header contains information about number of channels, frequency or wavelength, instrument's name, longitude/latitude of each pixel, integration time, gain factors, etc. *Raphael GCS* is compatible to read/write this kind of metadata written on hdf format. Figure 4.II.1 shows few metadata about satellite multispectral maps.

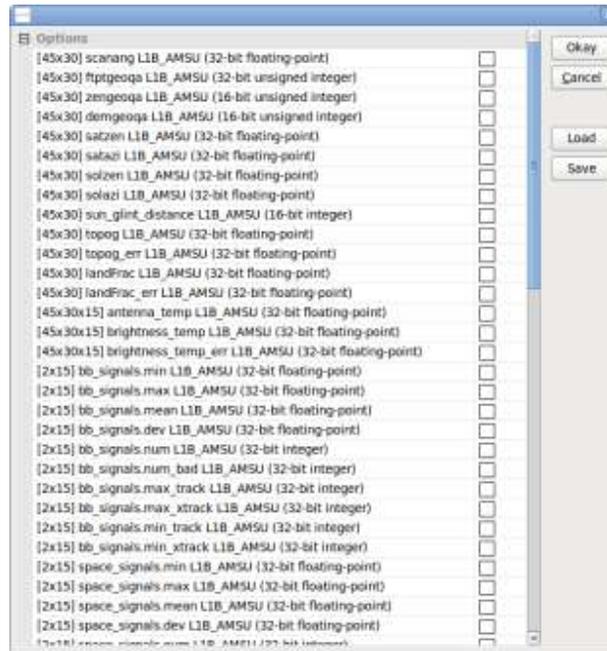


Figure 4.II.1: Metadata taken from the AIRS instrument version L1B.

In principle, this module can displays satellite metadata likes maps. The physical quantity that may contains this kind of metadata can be: Brightness Temperature, Antenna Temperature, Chemical Concentration of some substance, etc. The figure 4.II.2, shows two channels of AIRS metadata.

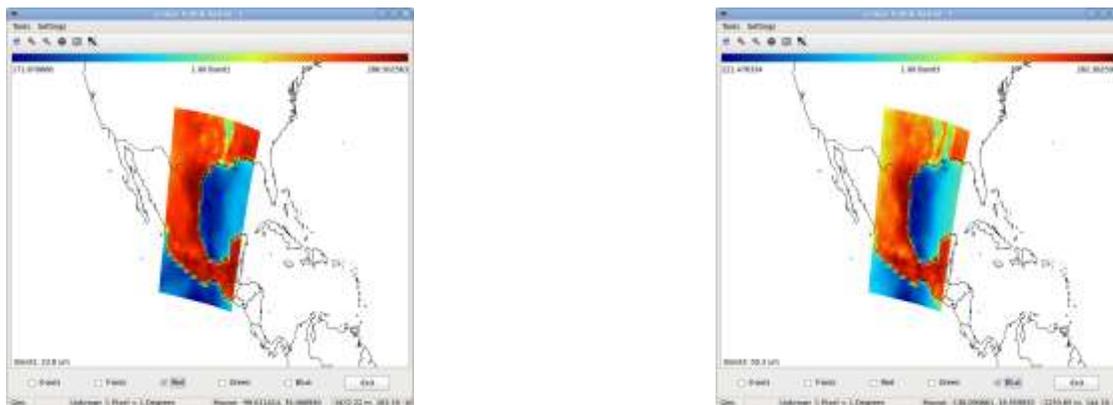


Figure 4.II.2: It is shows, Brightness Temperature maps. Left) Band 1 at $23.8 \mu\text{m}$, Right) Band 3 $50.3 \mu\text{m}$

The whole GIS operations defined in section 3.II can be apply on multispectral metadata. Particularly, algebraic operations user-defined on map space, as shown in figure 4.II.3.

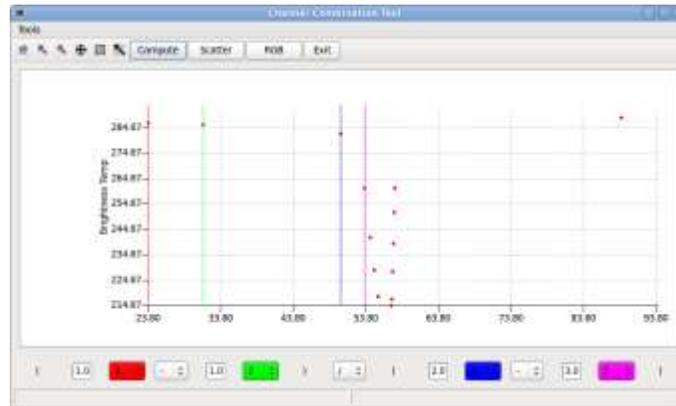


Figure 4.II.3: It is shows channel combination tool, which is enable to compute algebraic operations on the multispectral maps.

For example, algebraic operation like $Band\ 1 - Band\ 2$ or $Band\ 3 - Band\ 1$ can be performed. The result is shown in figure 4.II.4.

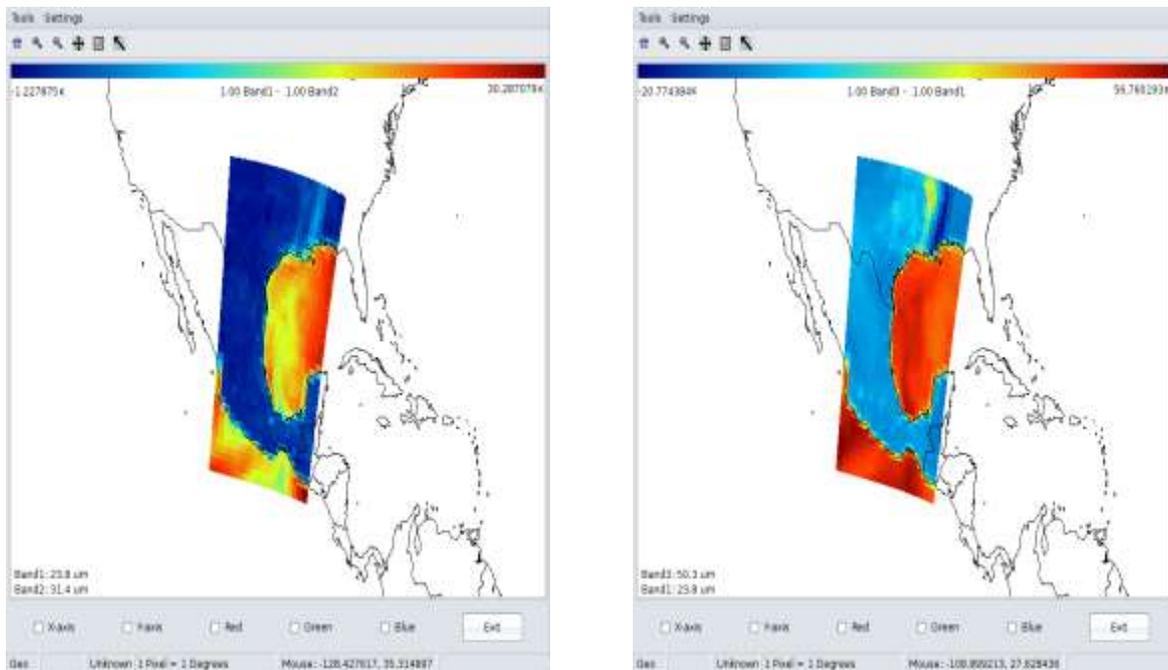


Figure 4.II.4: Two linear combination maps from AIRS data bands. Left) $Band\ 1 - Band\ 2$ and Right) $Band\ 3 - Band\ 1$.

However, we can design more complex calculations such as: $\frac{Band_1 - Band_2}{2\ Band_3 - 3\ Band_5}$.

Additionally, we can plot scatter data using the scatter section. It is useful when we have see if there is correlate/uncorrelate information between multiple channels. Furthermore, we can use principal component analysis to extract information at high-dimensional level (see section below). The figure 4.II.5 shows an example of scatter data and RGB combine task.

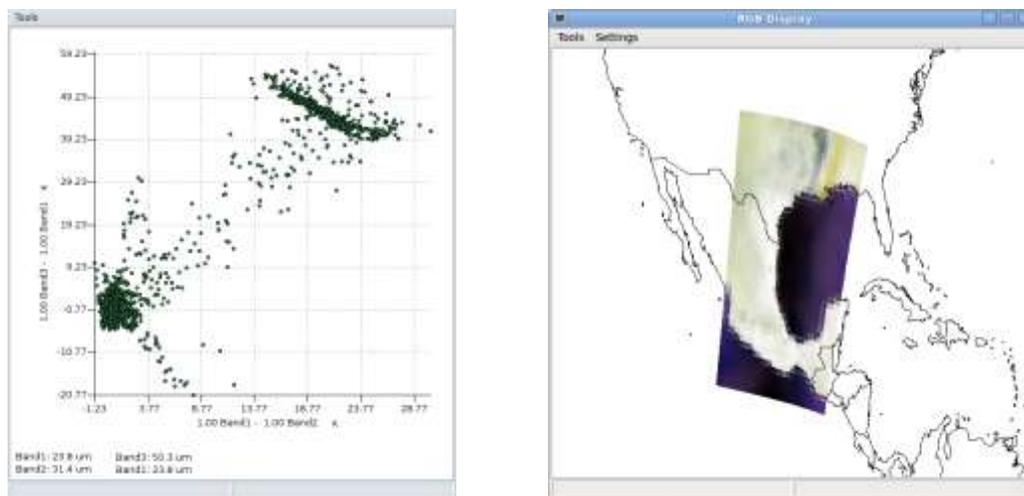


Figure 4.II.5: It is shows scatter plot of two satellite maps (left). It is shows a RGB combination of selected bands (right).

Remember that we can perform the whole GIS operations set, as defined in the Appendix. However, we mention a particular operation over multispectral metadata, such as: Filter to detect vegetation or Normalized Difference Vegetation Index (NDVI). *Raphael GCS* can calculate this kind of coefficients and all its variants. The figure 4.II.6 shows the NDVI index around Guadalajara City. Green colour mark the distribution of vegetation, while cyan colour corresponds to areas of rock, sand or snow.

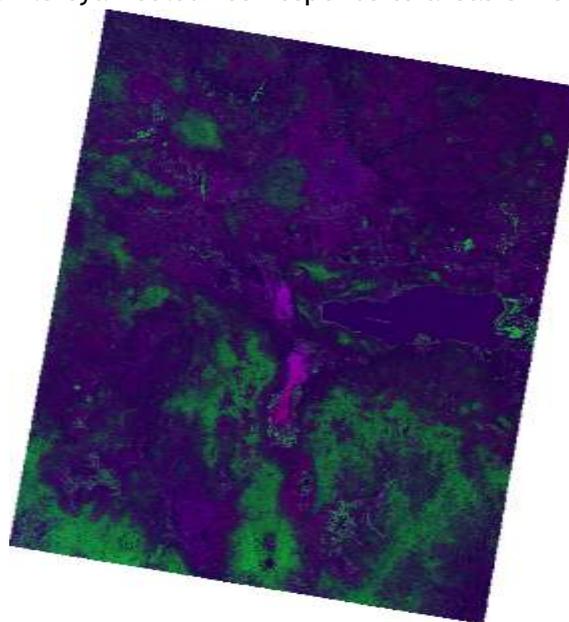


Figure 4.II.6: NDVI index around Guadalajara City, using Lansat-7 data.

Multivariate Methods.

Principal components analysis (PCA): It is a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for how much contribution in the data as possible, and each succeeding component accounts for as much of the remaining weight as possible. In this way, PCA is often used to reduce the number of variables, or dimensions, in a data set in order to simplify analysis or aid in visualization. PCA is for compressing a set of high dimensional vectors into a set of lower dimensional vectors and then reconstructing the original set. It is a non-parametric method of extracting relevant information from confusion data sets.

PCA has limitations that can be solved using other mathematical methods such as Independent Component Analysis (ICA). The assumption that normally PCA fails is when non-gaussian or multimodal gaussian are present on the data sets. This is not a guarantee that the directions of maximum variance will contain good features for discrimination.

Example:

Given some input data as shown in the figure 4.11.7. The experiment is given by five input data. The covariance matrix was calculated. Then, we get the eigenvector and eigenvalues. Finally, the principal components are calculated.

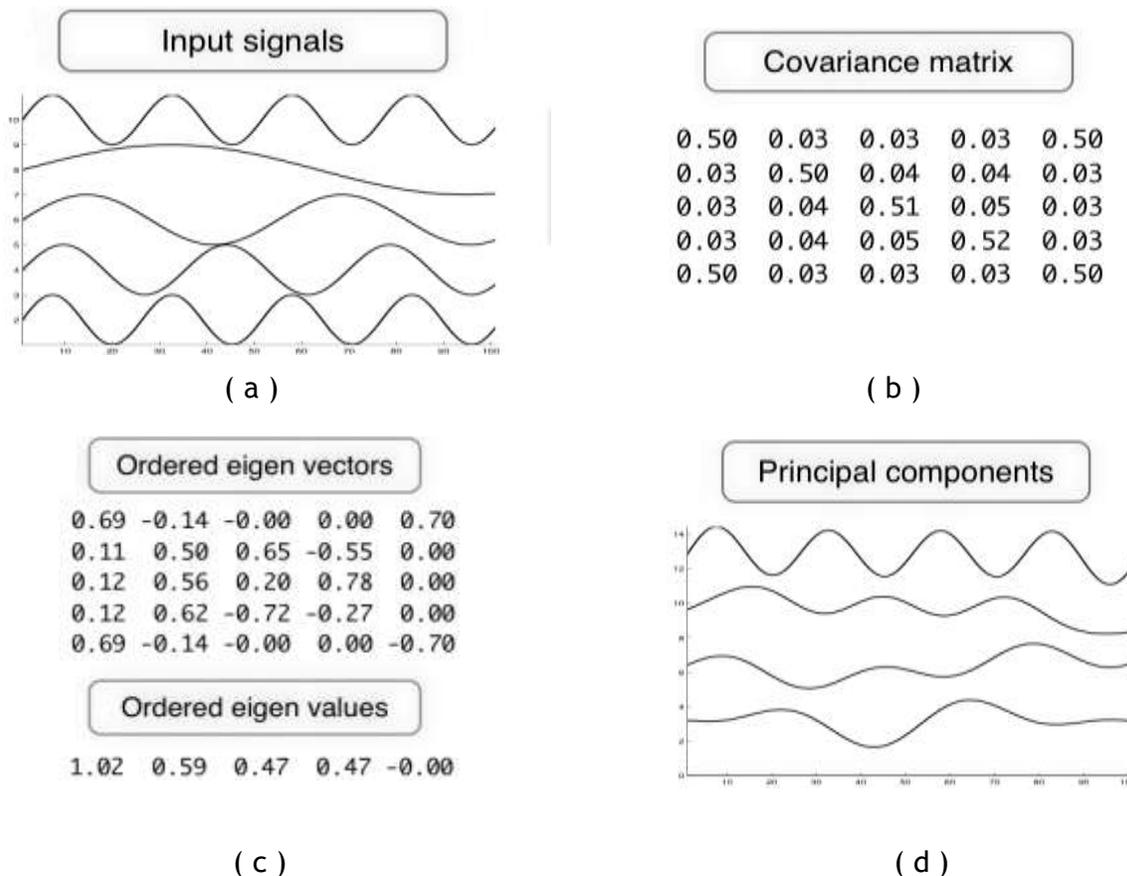


Figure 4.11.7: (a) Experimental input data, (b) Covariance matrix, (c) Eigenvectors and eigenvalues and (d) Principal components.

Independent Component Analysis (ICA): It is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals.

ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed nongaussian and mutually independent, and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA.

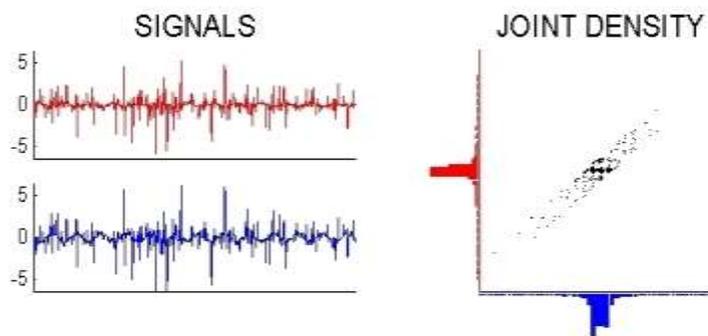
ICA is superficially related to principal component analysis and factor analysis. ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail completely.

The data analyzed by ICA could originate from many different kinds of application fields, including digital images, document databases, economic indicators and psychometric measurements. In many cases, the measurements are given as a set of parallel signals or time series; the term blind source separation is used to characterize this problem. Typical examples are mixtures of simultaneous speech signals that have been picked up by several microphones, brain waves recorded by multiple sensors, interfering radio signals arriving at a mobile phone, parallel time series obtained from some industrial process or multispectral satellite data image.

Example: Blind Source Separation (BSS) Problem.

Given two linear mixtures of two source signals which we know to be independent of each other, i.e. observing the value of one signal does not give any information about the value of the other. The BSS problem is then to determine the source signals given only the mixtures.

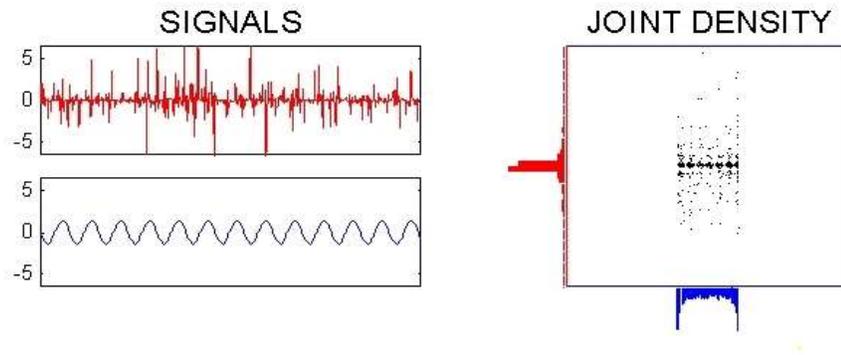
This figure 4.11.8 (below) shows the signal mixtures on the left and the corresponding joint density plot on the right. That is, at a given time instant, the value of the top signal is the first input component, and the value of the bottom signal is the corresponding second input component. The plot on the right is then simply plotting both signal along first and second input component. The marginal densities are also shown at the edge of the plot.



Input signals and density

Figure 4.11.8: Left) Input signals components (measured signals). Right) Joint density of both input components.

There are many algorithms for performing ICA, but the most efficient to date is the FastICA (fixed-point) algorithm which is implemented on *Raphael GCS*. The plot below shows the result after convergence of the FastICA algorithm.



Separated signals after 5 steps of FastICA

Figure 4.II.9: Result after FastICA is applied.

The source signals in this example were a sinusoid and impulsive noise, as can be seen in the left part of the plot below. The right plot shows the joint density which can be seen to be the product of the marginal densities, i.e. the definition of independence.

The following section is an example of applying PCA and ICA on satellite image data.

PCA method:

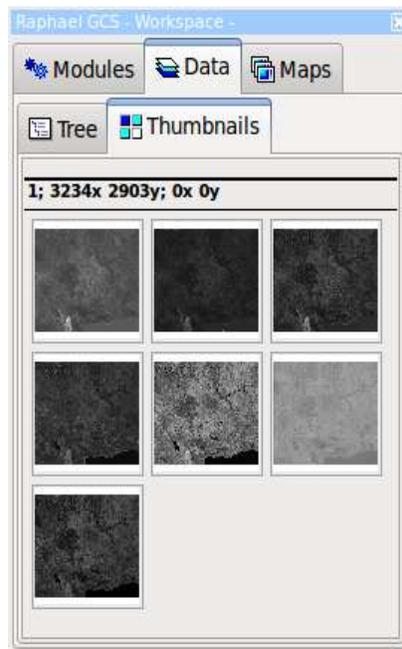


Figure 4.II.10: Lansat-7 as input data for PCA calculations.

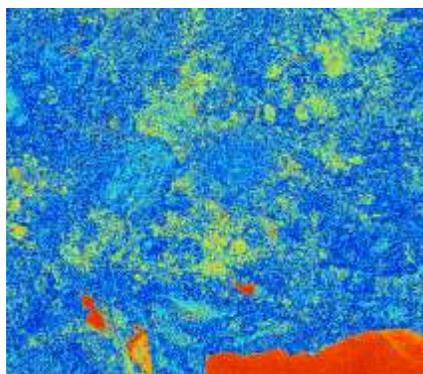
The covariance matrix and eigenvalues calculated are given by:

01. Map	02. Map	03. Map	04. Map	05. Map	06. Map	07. Map	Covariance Matrix
	1	2	3	4	5	6	7
1	456.865295	315.781097	600.248657	528.777222	1602.683105	622.828125	1011.075195
2	462.038818	353.405670	563.622437	420.210754	1087.736084	315.781097	706.635498
3	604.848877	420.210754	742.812073	1108.754150	1911.196289	528.777222	1107.785522
4	763.136902	462.038818	794.773621	604.848877	1563.328613	456.865295	1024.065796
5	794.773621	563.622437	1171.009521	742.812073	1960.635742	600.248657	1293.747803
6	1024.065796	706.635498	1293.747803	1107.785522	3501.788086	1011.075195	2547.533203
7	1563.328613	1087.736084	1960.635742	1911.196289	6981.224609	1602.683105	3501.788086

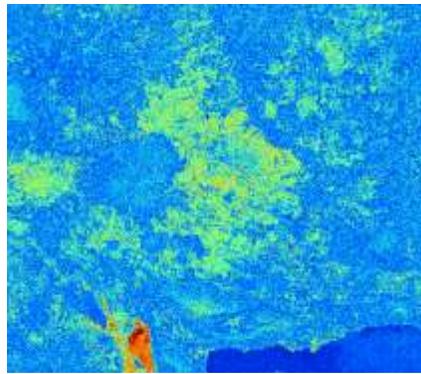
01. Map	02. Map	03. Map	04. Map	05. Map	06. Map	07. Map	Eigenvalues
	1	2	3	4	5	6	7
1	3046.861816	866.863525	493.075409	369.764099	144.417526	126.050346	34.527451

Figure 4.II.11: Shows covariance matrix and eigenvalues.

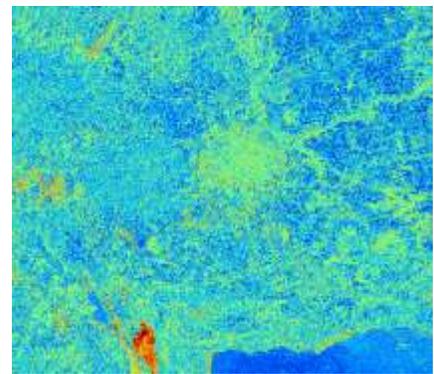
After applying PCA on this satellite data image set, we have the following principal components:



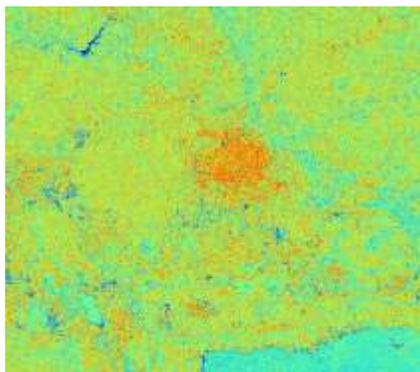
(a)



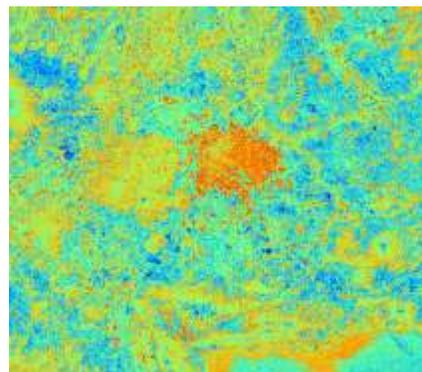
(b)



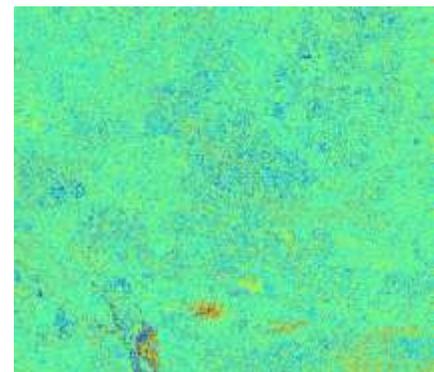
(c)



(d)



(e)



(f)

Figure 4.II.12: It is shows the six principal components of the satellite image data. (a) It is the first principal component. From (b) to (f) it is shows the decreasing contribution to the total signal image.

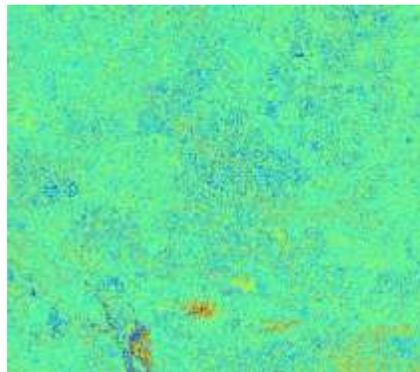


Figure 4.II.13: Noisy image component.

We can see that the first component (see figure 4.II.12 a) has strong contribution to the total image set, while last component (see figure 4.II.13) is associated with the nosier image component.

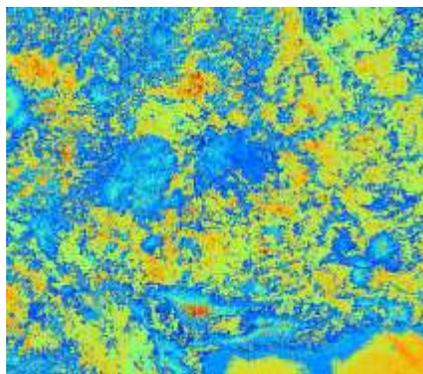
Fast ICA method.

Once the centering and whitening step is done and using the minimization function: $y = xe^{-x^2/2}$ is performed, we get the covariance matrix which is given by:

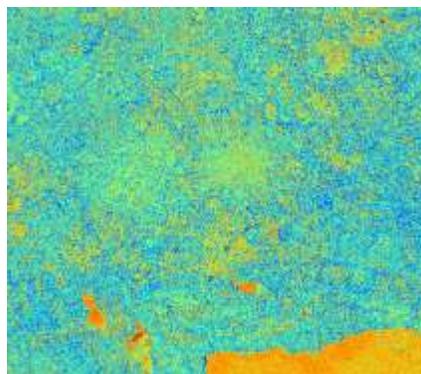
01. Map	02. Map	03. Map	04. Map	05. Map	06. Map	07. Map	Covariance Matrix		
	1	2	3	4	5	6	7		
1	763.136902	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.000000	353.405670	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000000	1171.009521	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000000	1108.754150	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000000	6981.224609	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000000	622.828125	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	2547.533203	0.000000

Figure 4.II.14: Shows covariance matrix.

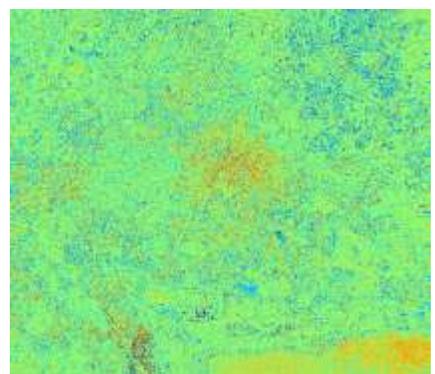
After, the convergence is reached, the principal components is given by the following images (see figure 4.II.15) :



(a)



(b)



(c)

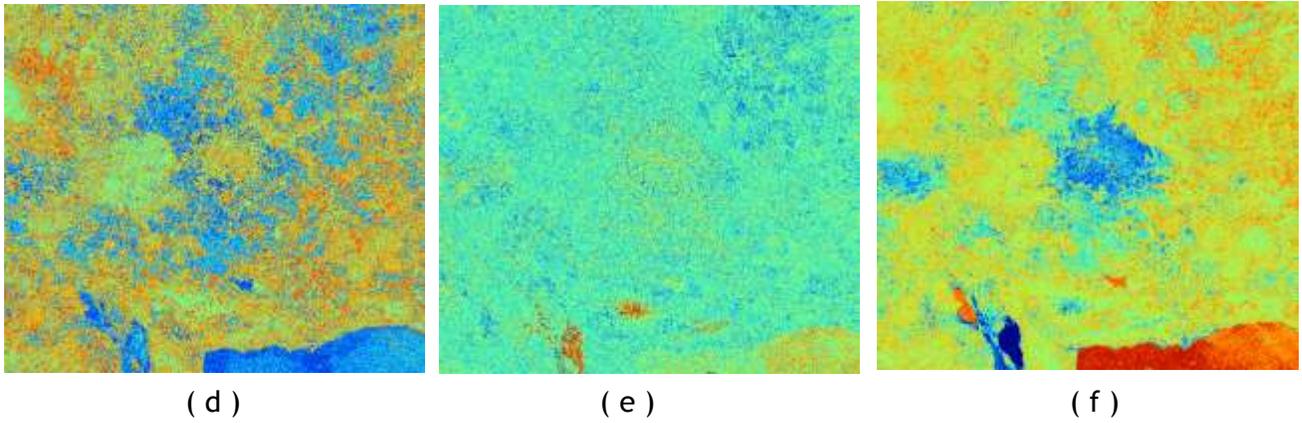


Figure 4.II.15: It is shown the principal components using the FastICA algorithm. (a) It is shows the first component. From (b) to (f) it is shows the descending contribution image to the original image data set.

Finally, the weakest component is give by figure 4.II.16.

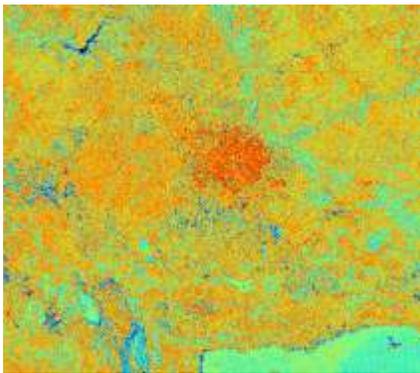


Figure 4.II.16: It shows the weakest component.

4.III SATELLITE DATA ACCESS

We currently have access to data from only two satellites through their instrument aboard, MODIS aboard Terra and AIRS aboard Aqua, in a format compatible with our system but we hope to provide access to further sources over the coming months. This satellite information is read/written using the standard files .hdf version 4 and 5. The multispectral module is fed through this satellite data access, too. In this way, UAV imaging can be overlaid on satellite data, thus providing fine detail combined with the global context of areas under study.

Satellite information is normally written using .hdf format version 4 or 5. *Raphael GCS* is compatible to read/write this kind of metadata, for example the following dialog window is from the hdf4 header of AIRS.

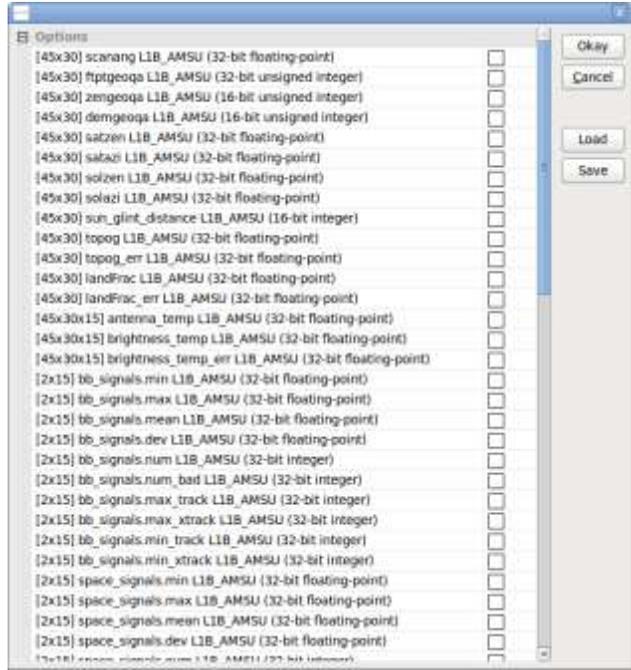


Figure 4.III.1: Header of the hdf file format of AIRS instrument aboard Aqua satellite.

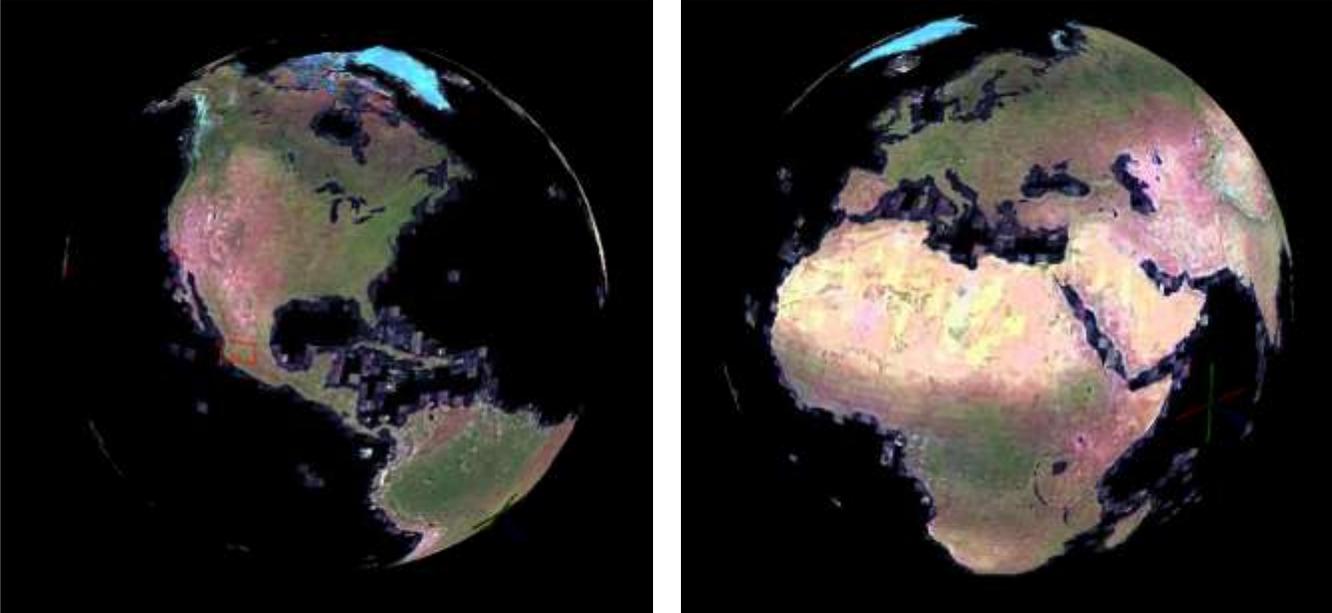


Figure 4.III.2: World map representation of Lansat-7 data. It uses our 3D visualization toolkit.

5. Standard "Raphael Ground Control Station"

The standard specification of the software as previously described can be combined with hardware into a standard package containing the following items:

- Basic *Raphael GCS* software suite
- Manual radio-control
- Telemetry transmitter/receiver
- Directional antenna mounted on a gimbal able to communicate with a vehicle at up to 30 Km. in direct line of sight.
- Ruggedized laptop
- User manual

6. Pricing details

The following are the launch costs associated with the *Raphael GCS* software, which compare very favourably with the costs of in-house development and/or using hardware such as a LIDAR for doing 3D reconstruction for instance. The cost of each module is as follows:

- Basic System: €6,500 for a site license irrespective of the number of users, and €500 per yearly update.
- FMV to 3D module: €3,500 for a site license irrespective of the number of users, including the CAD to 3D object comparison routine.
- Multi-spectral analysis: €2,500 for a site license irrespective of the number of users, including multivariate analysis and image fusion routines.
- Satellite access: €2,500 for a site license irrespective of the number of users, currently supported satellite sensors include AIRS, MODIS and LANDSAT.

7. Collaboration Structure

We believe that in order to establish *Raphael GCS* as the leading GCS in the market, it is important to accommodate the need of many companies of not only acquiring technology but also being able to add value by adding their own modules and further developments of sensors and other peripherals to the basic platform. Using a common GCS platform will help developments by implementing individual contributions (EO sensors, control algorithms, communications, propulsion, energy management, etc.) on a common interoperable system that can be used by or marketed to third parties. AVNTK can contribute by making available an advanced software product, which has taken us 4 years to develop, that will be cost-effective, reduce development costs, shorten time-to-market and reduce duplication of effort on a crucial area. In this context, we offer the following facilities to the various market participants:

UAV manufacturers: We are able to provide engineering services on an outsourcing basis to complement the customer software suites currently being used, with the system herein described. The customer needs to get a licence for using our software, and will have full ownership of any software done specifically for him. Where the software is sold on to a final customer, we will pay a 25% commission to the UAV manufacturer.

Engineering companies and UAV resellers: We would be prepared to share some of the software work with the collaborating company. Each entity would own their own developments. The customer needs to get a license for using our software, and will have full ownership of any software done specifically for him. The 25% commission would be payable to the collaborating company in the case of work done for a final customer, or shared with the UAV manufacturer if applicable.

UAV users: To help develop a collaborative application base is beneficial to everyone, as shown by cases such as the Neuron UCAV, Airbus, JSF, ISS... a single country/company cannot succeed! We are happy to support end-users both with technology and also with consultancy services, both to themselves and/or to their prime contractors or other suppliers, such as universities, R&D centres, etc.

Revisions history

Rev #	Date	Description	Revised by	Sections affected / Remarks
1.0	20/10/2009	Preliminary Draft	MFG	First draft
1.1	01/01/10	First Draft	FGR	

8. Appendices

8.1 DETAIL OF GIS FUNCTIONS:

The GIS module currently has basic operation with grids such as:

Elementary grid operations.

- **Function:** Generate a grid based on a function expression. The function interpreter uses an expression parser that offers the following operators:

Addition, Subtraction, Multiplication, Division, power, $\sin(a)$, $\cos(a)$, $\tan(a)$, $\text{asin}(a)$, $\text{acos}(a)$, $\text{atan}(a)$, $\text{atan2}(a,b)$, $\text{abs}(a)$, $\text{int}(a)$, $\text{sqrt}(a)$, $\text{mod}(a,b)$, $\text{gt}(a,b)$ (return 1 if a greater b), $\text{lt}(a,b)$ (returns 1 if a lower b), $\text{eq}(a,b)$ return 1 if a equal b, The variable are x and y. Examples: $\sin(x*x+y*y)(x*x+y*y)$,

- **Geometric Figures:** Construct grids from geometric figures (planes, cones).
- **Grid Calculator:** One of the most useful modules that you can find in *Raphael GCS* is the grid calculator. Since it is a very flexible module, it allows your to perform all kind of arithmetical operations with grids, and can be, therefore, used for a large variety of purposes. Operations such as: Addition, subtraction, multiplication, division, power, trigonometric functions, transcendental operations, boolean operations.
 - Changing grid orientation: *Raphael GCS* has its own module to do basic operations using grid, such as rotating, mirror, stretch, Flip or Invert an image.
- **Grid Normalisation:** This is quite common when you deal with several grids and you want to use a model in which each parameter should be scaled according to a particular scale such as the trivial range 0-1. You can normalize a grid using the grid calculator. If “standard deviation” method is selected, then the standard deviation of the resulting normalized grid will equal 1.
- **Grid Volume:** Calculate the volume under the grid's surface. This is mainly useful for Digital Elevation Models (DEM).
- **Random Field:** Create a grid with pseudo-random numbers as grid cell values.
- **Random Terrain Generation.**
- **Filtering grid:** A filter modifies all the cells in a grid (applying different algorithms and formulas) to generate a new version of that grid. The grid you obtain after a filtering process represents the same variable and in the same units as the original grid. The most common use for a filter is the elimination of noise. Although, the art of filter is widely used in detection circles, lines, or any other template. Single cells with unexpectedly high values are a good example of what noise means. When dealing with images instead of data grids, filters are a basic tool that is constantly used to prepare those images before any further processing. However, you should be careful when applying filters to data grids, since their effect can significantly alter their information. For example, applying a smoothing filter to a DEM is a somehow “rude” way of eliminating small sinks, but is not recommended, since it changes the height information contained in all of the cells, while only some of them should be modified. Although it removes noise, a smoothing filter causes the filtered grid to have a lower level of detail than the original one.

Always have this in mind when you use a filter: The less you modify the grid (assuming its information is reliable and comes from a good source), the more accurate your results will be.

There are several simple filters available on *Raphael GCS*, such as:

- Average
- Sharp
- Edge
- Neon
- Sobel
- Laplace
- Difference of Gaussian
- Gaussian: The degree of smoothing is determined by the standard deviation.
- Exponential
- Nonlinear
- Multidirectional Lee: This direction Lee Filter is an enhanced Lee Filter. It looks into 16 directions for the direction with the minimum variance and applied a Lee Filter on this direction. Uses this filter for remove speckle noise in SAR images or DTMs. On DTMs this filters will preserve the slope and narrow valleys.
- User defined Filter

Interpolation of grids.

- **B-Spline Approximation:** Calculates B-spline functions for chosen level of detail. This module serves as basis for the Multilevel B-Spline Interpolation' and is not suited as it is for spatial data interpolation from scattered data.
- **Multilevel B-Spline Interpolation:** Multilevel B-spline algorithm for spatial interpolation of scattered data. The algorithm makes use of a coarse-to-fine hierarchy of control lattices to generate a sequence of bicubic B-spline functions, whose sum approaches the desired interpolation function. Large performance gains are realized by B-spline refinement to reduce the sum of these functions into one equivalent B-spline function. The maximum level determines the maximum size of the final B-spline matrix and increases exponential with each level. Where level = 10 requires about 1MB, level = 12 needs about 16MB and level = 14 about 256MB of additional memory.
- **Multilevel B-Spline Interpolation (from grid).**
- **Thin Plate Spline (Global).**
- **Thin Plate Spline (Local).**
- **Thin Plate Spline (TIN).**

Geostatistics

Kriging is a complex geostatistical technique used to create regular grids from irregularly spaced data points. There are several Variogram Model to fit the data, such as: Spherical, Exponential, Gaussian, Linear Regression, Exponential Regression, Power Function Regression.

- **Ordinary Kriging:** Ordinary Kriging for grid interpolation from irregular samples points.
- **Ordinary Kriging (Global):** This implementation does not use a maximum search radius. The weighting matrix is generated once globally for all points.
- **Ordinary Kriging (VF):** Kriging with Variogram Fit.

- **Ordinary Kriging (VF, Global):** Kriging with Variogram Fit. The weighting matrix is generated once globally for all points.
- **Universal Kriging.**
- **Universal Kriging (Global).**
- **Universal Kriging (VF).**
- **Universal Kriging (VF, Global).**
- **Semi-Varigram (Points).**
- **Point Analysis:** Ripley's K, variation ellipse, etc.
- **Multiple regression analysis:** Linear regression analysis of point attributes with multiples grids. Details of the regression/correlation analysis will be saved to a table. The regression functions is used to create a new grid with (extrapolated) values. The multiple regression analysis uses forward selection procedure.
- **Radius of variance:** Find the radius within which the cell values exceed the given variance criterium. This modules is closely related to the representativeness calculation (variance within given search radius). For easier usage, the variance criterium is entered as standard deviation value.
- **Regression analysis:** Regression analysis of point attributes with grid values. The regression function is used to create a new grid with (extrapolated) values.
- **Representativeness:** Calculation of the variance within a given search radius.
- **Residual analysis:** Relations of each grid cell to its neighborhood. This calculations can be used on terrain analysis.
- **Statistics for grids:** Calculates statistical properties (arithmetic mean, minimum, maximum, variance, standard deviation) for each cell position for the values of the selected grids.
- **Zonal grid statistics:** The module can be used to create a contingency table of unique condition units (UCUs). These units are delineated from a zonal grid (eg. Sub catchments) and optional categorical grids (eg. Lancover, soil,...) It is posible to calculate simple statistics (min, max, mean, standard deviation and sum) for each UCU from optional grids with continuous data (eg. Slope). The number of input grids is only limited by available memory. The module has four different modes of applications: 1) only a zonal grid is used as input. This results is a simple contingency table with the number of grid cells in each zone. 2) a zonal grid and additional categorical grids are used as input. This results in a contingency table with the number of cells in each UCU. 3) a zonal grid and additional grids with continuous data are used as input. This results in a contingency table with the number of cells in each zone and some simple statistics for each zone. The statistics are calculated for each continuous grid. (4) a zonal grid, additional categorical grids and additional grids with continuous data are used as input. This results in a contingency table with the number of cells in each UCU and the corresponding statistics for each continuous grid.

Discretization

This is a tool for the discretization / classification of metric gridded data.

- **Cluster analysis for grids:** An approach for solving the taxonomy problem.
- **Grid segmentation:** Segmentation with the local maximum method.
- **Grid segmentation (Option B):** Segmentation with the local minimu/maximum method.
- **Grid skeletonization:** Skeletonisation methods: Stadard, Hilditch, Channel Skeleton.
- **Supervised classification:** Minumum distance, Maximum Likelihood.

Gridding

Tools for the gridding of points and other vector data.

- **Inverse Distance:** Inverse distance to a power method for grid interpolation from irregular distributed points.
- **Modified Quadratic Shepard:** Modified quadratic shepard method for grid interpolation from irregular distributed points.
- **Nearest Neighbor:** Natural neighbor method for grid interpolation from irregular distributed points.
- **Shapes to Grid:** Gridding of a shapes layer. If some shapes are selected, only these will be gridded.
- **Triangulation:** Gridding of a shapes layer using Delaunay Triangulation.

Advanced grid operations I

These modules are for edit, delete, mix, fill, overlap, create manipulation of gridded data.

- **Aggregate:** For a given i,j cell assign max/min/sum within a given window of radius r.
- **Change Grid Values:** Changes values of a grid according to the rules of a user defined lookup table. Values of value ranges that are not listed in the lookup table remain unchanged. If the target is not set, the changes will be stored to the original grid.
- **Change Cell Values [Interactive].**
- **Change Grid Values - Flood Fill [Interactive]:** Interactively use the flood fill method to replace a grid's cell values. If the target is not set, the changes will be stored to the original grid.
- **Close Gaps:** Close gaps of a grid data set (i.e. Eliminate no data values). If the target is not set, the changes will be stored to the original grid.
- **Close One Cell Gaps:** Close one cell gaps using the mean value of the surrounding cell values. If the target is not set, the changes will be stored to the original grid.
- **Combine Grid:** Combine two given grids.
- **Convert Data Storage Type:** Changes the storage data type of a grid (e.g. From 4 byte floating point to 2 byte signed integer). If the target is not set, the original grid's storage type will be changed.
- **Create Constant Grid.**
- **Create Grid System:** This module creates a new user Grid System for use with other modules.
- **Crop to Data:** Crop grids to valid data cells.
- **Cutting [Interactive]:** Create a new grid from interactively selected cut of an input grid.
- **Grid Buffer:** Grid buffer creation.
- **Grid Orientation:** Copy, mirror (horizontally or vertically) and invert grid values. If the target is not set, the changes will be stored to the original grid.
- **Grid Proximity Buffer:** This module calculates the euclidean distance within a buffer distance from all NoData cells to the nearest valid neighbor in a source grid. Additionally, the source cells define the zones that will be used in the euclidean allocation calculations. Cell values in the source grid are treated as Ids (integer) and used in the allocation grid to identify the grid value of the closest source cell. If a cell is at an equal distance to two or more sources, the cell is assigned to the source that is first encountered in the modules scanning process. The buffer grid is reclassification of the distance grid using a user specified equidistance to create a set of discrete distance buffers from sources features. The buffer zones are coded with the maximum distance grid is floating-point. The output values for the allocation and buffer grid are of type

integer. The duration of module execution is dependent on the number of sources cells and the buffer distance.

- **Grid Value Request [Interactive]:** Grid value Request. It uses an interpolation method.
- **Grids from classified grid and table:** Creates several grids using a classified grid and a table with data values for each class.
- **Invert Data/No-Data.**
- **Merging.**
- **Patching:** Fill gaps of a grid with data from another grid.
- **Reclassify Grid Values:** The module can be used to reclassify the values of a grid. It provides three different options: a) reclassification of single values, b) reclassification of a range of values and c) reclassification of value ranges specified in a lookup table. In addition to these methods, two special cases (NoData values and values not included in the reclassification setup) are supported. In mode (a) and (b) the “NoData option” is evaluated before the method settings, in mode (c) the option is evaluated only if the NoData values are not included in the lookup table. The “other values” option is always evaluated after checking the method settings.
- **Resampling:** Resampling of grids.
- **Sort Grid.**
- **Threshold Buffer:** Threshold buffer creation.

Advanced grid operations II

- **Accumulated Cost (Anisotropic).**
- **Accumulated Cost (Isotropic).**
- **Aggregation Index:** It is useful to quantify spatial patterns of landscapes.
- **Analytical Hierarchy Process.**
- **Change Vector Analysis.**
- **Covered Distance.**
- **Cross-Classification and Tabulation.**
- **Fragmentation (Alternative):** 1) interior, if $Density = 1.0$, 2) undetermined, if $Density > 0.6$ and $Density = Connectivity$, 3) perforated, if $Density > 0.6$ and $Density - Connectivity > 0$, 4) edge, if $Density > 0.6$ and $Density - Connectivity$, 5) transitional, if 0.4 , 6) patch, if $Density$.
- **Fragmentation (Standard):** Grid fragmentation analysis given by Riitters method.
- **Fragmentation classes from density and connectivity.**
- **Fuzzify:** Prepares a grid for fuzzy logic analysis.
- **Fuzzy intersection grid:** Calculates the intersection for each grid cell of the selected grids.
- **Fuzzy union grid:** Calculates the union for each grid cell of the selected grids.
- **Layer of extreme value:** It creates a new grid containing the ID of the grid with the maximum (minimum) value.
- **Least cost path (interactive):** Creates a least cost path profile using an accumulated cost surface.
- **Ordered weighted averaging (OWA):** Ordered Weighted Averaging (OWA).
- **Pattern analysis.**
- **Polar to Rect:** Polar to Rect conversion for paired vector data.
- **Rect to Polar:** Rect to Polar conversion for paired vector data.
- **Soil texture classification:** Define soil texture with USDA scheme from sand and clay contents.

Classification	
Clay soils	Silt soils
SiltyClay soils	SiltLoam soils
SiltyClayLoam soils	Loam soils
SandyClay soils	Sand soils
SandyClayLoam soils	LoamySand soils
ClayLoam soils	SandyLoam

Table 3.II.1: Soil texture classification.

- **Vegetation Index (distance based):** Distance-based vegetation indexes, 1) Richardson-Wiegand (PVI), 2) Perry and Lautenschlager), 3) Walter and Shabaani, 4) Qi.
- **Vegetation Index (slope based):** Slope-based vegetation indexes. 1) Normalized difference vegetation, 2) Ratio vegetation index, 3) Transformed vegetation index, 4) Corrected transformed vegetation, 4) Thiam's transformed vegetation, 5) Normalized ratio vegetation index.

Import and Export between raster/vector data formats.

Import and Export of Data: Fundamental for the work with spatial data are interfaces to the countless data file formats. Particularly the data exchange between different programs usually requires a suite of import and export filters. *Raphael GCS* offers several filters to common data formats, including various image. Most flexible is a raster data import tool, that uses the Geospatial Data Abstraction Library (GDAL), which alone supports more than 40 different file formats.

- **GPSBabel:** An interface to the GPSBabel software. GPSBabel converts waypoints, tracks, and routes between popular GPS receivers and mapping programs. It also has powerful manipulation tools for such data.

Geocaching.com .loc	GPSman	GPX XML	Magellan protocol
Magellan Mapsend	Garmin PCX5	Garmin Mapsource	gpsutil
U.S. Census Bureau Tiger Mapping Service	Comma separated values	Delorme Topo USA4/XMap Conduit	Navitrak DNA marker format
MS PocketStreets 2002 Pushpin	Cetus for Pals/OS	GPSPilot Tracker for Palm/OS	Magellan NAV companion for PalmOS
Garmin serial protocol	MapTech Exchange Format	Holoux (gm-100) .wpo Format	OziExplorer Waypoint
National Geographic Topo .tpg	TopoMapPro Places File		

Table 3.II.2: Available formats for PGS Babel interface.

- **GPX to shapefile:** Converts a GPX file into a shapes (.shp).
- **Import DXF Files:** This module imports DXF files using the free “dxflib” library.
- **Import ESRI E00 File:** Import data sets from ESRI's E00 interchange format. This import filter is based on the E00 format analysis of the GRASS GIS module 'm.in.e00'.

- **Import GRIB2 record:** Import a GRIB2 record.

Geospatial Data Abstraction Library

Raphael GCS has implemented an interface to import and export the data formats available through the GDAL library.

- **Export Raster:** The “GDAL Raster Export” module exports one or more GIS to various file formats using the “Geospatial Data Abstraction Library” (GDAL).

Currently raster supported formats are:

ID	Name	ID	Name
VRT	Virtual Raster	ERS	ERMapper (.ers)
GTiff	GeoTIFF	RMF	Raster Matrix Format
NIFT	National Imagery Transmission Format	RST	Idrisi Raster A.1
HFA	Erdas Imagine Images (.img)	INGR	Intergraph Raster
ELAS	ELAS	GSBG	Golden Software Binary Grid (.grd)
MEM	In Memory Raster	PNM	Portable Pixmap Format
BMP	MS Windows Device Independent Bitmap	ENVI	ENVI (.hdr)

Table 3.11.3: Export raster support formats.

PCDISK	PCDISK Database File	EHdr	ESRI (.hdr)
ILWIS	ILWIS Raster Map	PAux	PCI (.aux)
SGI	SGI Image File Format 1.0	MFF	Vexcel MFF Raster
Leveller	Leveller Heightfield	MFF2	Vexcel MFF2 (HKV) Raster
Terragen	Terragen Heightfield	BT	VTP (.bt) Binary Terrain 1.3 Format
HDF4Image	HDF4 Dataset	IDA	Image Data and Analysis
ADRG	ARC Digitized Raster Graphics		

Table 3.11.3: Export raster support formats. Continue.

- **Import Raster:** This module imports grid data from various file formats using the GDAL library.

Currently raster supported formats are:

ID	Name	ID	Name	ID	Name
VRT	Virtual Raster	ELAS	ELAS	GIF	Graphics Interchange Format (.gif)
GTiff	GeoTIFF	AIG	Arc/Info Binary Grid	BIGGIF	Graphics Interchange Format (.gif)
NIFT	National Imagery Transmission Format	AAIGrid	Arc/Info ASCII Grid	ESAT	Envisat Image Format

RPFTOC	Raster Product Format (.toc)	SDTS	SDTS Raster	BSB	Maptech BSB Nautilus Charts.
HFA	Erdas Imagine Images (.img)	DTED	DTED Elevation Raster	XPM	X11 PixMap Format
SAR_CEOS	CEOS SAR Image	PNG	Portable Network Graphics	BMP	MS Windows Device Independent Bitmap
CEOS	CEOS Image	JPEG	JPEG JFIF	DIMAP	SPOT DIMAP
JAXAPALSAR	JAXA PALSAR Product Reader (level 1.1/1.5)	MEM	In Memory Raster	AirSAR	AirSAR Polarimetric Image
GFF	Ground-based SAR applications Testbed File Format (.gff)	JDEM	Japanese DEM (.mem)	RS2	RadarSat 2 XML Product
PCIDSK	PCIDSK Database File	PCRaster	PCRaster Raster File	ILWIS	ILWIS Raster Map
SGI SGI Image File Format 1.0	SRTMHGT File Format	Leveller	Leveller heightfield	Terragen	Terragen heightfield

Table 3.II.4: Import raster support formats.

ID	Name	ID	Name	ID	Name
HDF4	Hierarchical Data Format Release 4	HDF4Image	HDF4 Dataset	ISIS3	USGS Astrogeology ISIS cube (version 3)
ISIS2	USGS Astrogeology ISIS cube (version 2)	PDS	NASA Planetary Data System	ERS	ERMMapper (.ers)
L1B	NOAA Polar Orbiter Level 1b Data Set	FIT	FIT image	GRIB	GRIdded Binary (.grb)
RMF	Raster Matrix Format	WCS	OGC Web Coverage Service	WMS	OGC Web Map Service
MSGN	EUMESAT Archive (.nat)	RST	Idrisi Raster A.1	INGR	Intergraph Raster
GSAG	Golden Software ASCII Grid (.grd)	GSBG	Golden Software Binary Grid (.grd)	GS7BG	Golden Software 7 Binary Grid (.grd)
COSAR	COSAR Annotated Binary Matrix (TerraSAR-X)	TSX	TerraSAR-X Product	COASP	DRDC COASP SAR Processor Raster
PNM	Portable PixMap Format (netpbm)	D0Q1	USGS D0Q (old style)	D0Q2	USGS D0Q (new style)
ENVI	ENVI (.hdr)	EHDR	ESRI (.hdr)	GenBin	Generic Binary (.hdr)
PAux	PCI (.aux)	MFF	Vexcel MFF Raster	MFF2	Vexcel MFF2 (HKV) Raster
FujiBAS	Fuji BAS Scanner Image	GSC	GSC Geogrid	FAST	EOSAT FAST Format
BT	VTP (.bt) Binary Terrain 1.3	LAN	Erdas (.lan, .gis)	CPG	Convair PolGASP
IDA	Image Data and Analysis	NDF	NLAPS Data Format	EIR	Erdas Imagine Raw
DIPEX	DIPEX	LCP	FARSITE v.4 Lascap File (.lcp)	RIK	Swedish Grid RIK (.rik)

USGSDEM	USGS Operational ASCII DEM (and CDED)	GXF	GeoSoft Grid Exchange Format	HTTP	HTTP Fetching Wrapper
HDF5	Hierarchical Data Format Release 5	HDFImage	HDF5 Dataset	ADRG	ARC Digitized Raster Graphics
BLX	Magallan topo (.blx)				

Table 3.II.4: Import raster support formats, continued.

- **Export Raster to GeoTiff via GDAL:** This module exports one or more grids to a Geocoded Tagged Image File Format using the “Geospatial Data Abstraction Library” (GDAL).
- **OGR - Export vector data:** This module exports vector data to various file/database formats using the GDAL library.

Currently vector data supported formats are:

Name	Name	Name	Name
ESRI Shapefile	MapInfo File	UK .NTF	SDTS
TIGER	S57	DGN	VRT
REC	Memory	BNA	CSV
GML	GPX	KML	GeoJSON
GMT	XPlane	AVCbin	AVCE00
Geoconcept			

Table 3.II.5: Export vector data.

- **OGR - Import vector data:** This module imports vector data from various file/database formats using the GDAL library (see Table. X).

Import/Export Grids

Tools for the import and export of gridded data currently available on *Raphael GCS*.

- **Export ESRI Arc/Info Grid:** Export grid to ESRI's Arc/Info grid format. Binary or ASCII format.
- **Export Grid to XYZ:** Export grid to a table (text format), that contains for each grid cell the x/y-coordinates and additionally data from selected grids.
- **Export Surfer Grid:** Export grid to Golden Software's Surfer grid format.
- **Export True Color Bitmap:** Export red-green-blue coded image grids to MS-Windows true color bitmaps. This module writes the data directly to the file and is hence particularly suitable for very large data sets.
- **Import Binary Raw Data:** Imports grid from binary raw data.
- **Import ESRI Arc/Info Grid:** Import grid from ESRI's Arc/Info grid format.
- **Import Erdas LAN/GIS:** Import Erdas LAN/GIS files.
- **Import Grid from Table:** Imports grid from table.
- **Import Grid from XYZ:** Import grid from a table (text format), that contains for each grid cell the x/y/z-coordinates and additional data from selected grids.

- **Import MOLA Grid (MEGDR):** Import Maras Orbit Laser Altimer (MOLA) grids of the Mars Global Survey (MGS) Mission (Topographic maps, Mission Experiment Gridded Data Records -MEGDRs).
- **Import SRTM30 DEM:** Extracts elevation grids from SRTM30 data.
- **Import Surfer Grid:** Import grid from Golden Software's Surface grid format.
- **Import USGS SRTM Grid:** Import grid from USGS SRTM (Shuttle Radar Topographic Mission) data.

Image Import and Export

- **Export Image:** Saves a grid as image using display properties as used by the graphical user interface (bmp, jpg, png).
- **Import Image:** Loads an image in formats bmp, jpg, png, tif, tiff, pnm, xpm.

Import and Export of vector data

- **Export Atlas Boundary File.**
- **Export GPX:** Exports vector data points to GPS eXchange format GPX.
- **Export GStat Shapes:** GStat shapes format export.
- **Export Shapes to Generate:** Export generate shapes format.
- **Export Shapes to XYZ:** XYZ export filter for shapes.
- **Export Surfer Blanking File:** Export shapes to Golden Software's Surfer Blanking File format.
- **Export TIN to Stereo Lithography File (STL):** An StL “StereoLithography” file is a triangular representation of a 3-dimensional surface geometry. The surface is tessellated or broken down logically into a series of small triangles (facets). Each facet is described by a perpendicular direction and three points representing the vertices (corners) of the triangle. These data are used by a slicing algorithm to determine the cross sections of the 3-dimensional shape to be built by the fabber. A fabber, short for digital fabricator.
- **Export WASP terrain map file.**
- **Import Atlas Boundary File.**
- **Import GPX:** Import GPS data from GPS eXchange format GPX.
- **Import GStat Shapes:** Gstat shapes format import.
- **Import Shapes from XYZ:** Point shapes import from text formatted XYZ-table.
- **Import Stereo Lithography File (STL).**
- **Import Surfer Blanking Files:** Import polygons/polylines from Golden Software's Surfer Blanking File format.
- **Import WASP terrain map File.**
- **Point Cloud from File.**

Tools for the import and export of tables.

- **Export Text Table.**
- **Import Text Table.**
- **Import Text Table with Number only.**

Shapes - Tools related to gridded and vector data

- **Add Grid Values to Points:** Retrieves information from the selected grids at the positions of the points of the selected points layer and adds it to the resulting layer.
- **Clip Grid with Polygon:** Clips the input grid with a polygon shapefile. Select polygons from the shapefile prior to module execution in case you like to use only a subset from the shapefile for clipping.
- **Contour Lines from Grid:** Create contour lines (isolines) from grid values.
- **Get Grid Data for Shapes:** Retrieves information from the selected grids at the positions of the shapes of the selected shapes layers and adds it to the resulting shapes layer.
- **Gradient from Grid:** Create lines indicating the gradient.
- **Grid Statistics for Polygons:** For each polygon statistics of the contained grid values will be generated.
- **Grid Values to Points:** This module saves grid values to a point shape. Optionally only points can be saved, which are contained by polygons of the specified shapes layer. In addition, it is possible to exclude all cells that are coded NoData in the first grid to the grid list.
- **Grid Values to Points (randomly):** Extract randomly points from gridded data.
- **Local Minima and Maxima:** Extracts local grid value minima and maxima of to vector points.
- **Vectorising Grid Classes:** Vectorising grid classes.

Shapes - Tools for lines.

- **Convert Points to Line.**
- **Convert Polygons to Lines:** Converts polygons shapes to lines shapes.
- **Line Properties:** Line properties such as length, number of vertices.

Shapes - Tools for the manipulation of point vector data.

- **Add Coordinates to Points.**
- **Clip Points with Polygons.**
- **Convert a Table to Points:** Create Point Theme From Table.
- **Count Points in Polygons.**
- **Create Point Grid:** Creates a regular grid of points.
- **Distance Matrix:** Computes distance between pairs of points in a layer.
- **Fit N Points to Shape.**
- **Points from Lines:** Converts a line theme to a points theme. Optionally inserts additional points in user-defined distances.
- **Remove Duplicate Points.**
- **Separates Points by Direction:** Separate points by direction. Direction is determined as average direction of three consecutive points A, B and C. If the angle between the directions of A-B and B-C is higher than given tolerance angle the point is dropped. This module has been designed to separate GPS tracks recorded by tractors while preparing a field.

Shapes - Tools for Polygons

- **Convert Lines to Polygons:** Converts lines shapes to polygon shapes. Line arcs are closed to polygons simple by connecting the last point with the first.
- **Convert Polygon/Line Vertices to Points.**
- **Geometrical Properties of Polygons.**
- **Polygon Centroids:**Creates a points layer containing the centroids of the input polygon layer.
- **Polygon Intersection:** Intersection of polygon layers. Uses General Polygon Clipper version 2.3.
- **Polygon Shape Indices:** Various indices describing the shapes of polygons. Based on area, perimeter, maximum distance between the vertices of a polygon.
- **Polygon Union:** The union of polygons, which have the same attribute value.
- **Polygon Statistics from Points:** Calculates statistics using a polygon and a points layer.

Shapes - Tools for the manipulation of vector data.

- **Create Chart Layer (bars/sectors).**
- **Create Empty Shapes Layer:** Creates a new empty shapes layer of given type, which might be either point, multipoint, line or polygon. Available field types for the attributes table are: string, 1, 2 and 4 byte inter, 4 and 8 floating point.
- **Create PDF Report for Shapes Layer.**
- **Create Web Content [Interactive].**
- **Create graticule.**
- **Cut Shapes Layer.**
- **Cut Shapes Layer [Interactive].**
- **Get Shapes Extents.**
- **Join a Table:** Joins a table with shapes layer's attributes.
- **Merge Shapes Layers.**
- **New Layer from Selected Shapes.**
- **QuadTree Structure to Shapes.**
- **Query Builder for Shapes.**
- **Search in Attributes Tables:** Searches for an expression in the attributes table and selects records where the expression is found.
- **Select by Theme:** Currently both input layers have to be of type polygon.
- **Separate Shapes.**
- **Shapes Buffer:** A vector based buffer construction partly based on the method supported on "An effective buffer generation method in GIS".
- **Split Shapes Layer.**
- **Split Shapes Layer Randomly:** Randomly splits one shapes layer into two new shapes layers. Useful to create a control group for model testing.
- **Split Table/Shapes by Attribute.**
- **Summary.**
- **Transform Shapes:** Use this module to move, rotate and/or scale shapes.

Table - This module is designed for table calculations.

- **Function Fit:** Calculates the least-square fit.
- **Running Average.**

- **Table Calculator:** The table calculator creates a new column based on the existing columns and a mathematical formula. The columns are addressed by single characters (a-z) which correspond in alphabetical order to the columns (a = first, b = second columns, etc).
- **Table Calculator for Shapes:** Calculation of the new values in attributes tables.
- **Trend for Shapes Data.**
- **Trend for Table Data.**

Table - Tools

- **Create Empty Table:** Creates a new empty table.
- **Enumerate a Table Attribute:** Enumerate attribute of a table, i.e. Assign to identical values of chosen attribute field unique identifiers.
- **Rotate Table:** Rotate a table, i.e. Swap rows and columns.

TIN - Tools for triangular Irregular Network (TIN) processing.

- **Flow Accumulation (Parallel):** Calculates the catchment area based on the selected elevation values.
- **Flow Accumulation (Trace):** Calculates the catchment area based on the selected elevation values.
- **Gradient:** Calculates the gradient based on the values of each triangle's points.
- **Grid to TIN:** Creates a TIN from grid points. NoData values will be ignored.
- **Grid to TIN (Surface Specific Points):** Creates a TIN by identifying (surface) specific points of a grid.
- **Shapes to TIN:** Convert a shapes layer to a TIN.
- **TIN to Shapes:** Convert a TIN data set to shapes layers.

Point Cloud - Tools for points clouds.

- **Point Cloud Cutter**
- **Point Cloud Cutter [interactive]**
- **Point Cloud from Grid Points**
- **Point Cloud from Shapes**
- **Point Cloud to Grid**
- **Point Cloud to Shapes**

Point Cloud - Point Cloud Viewer

- **Point Cloud Viewer**

Projection - Tools for coordinate transformation based on PROJ.4

Projections: Once data have been imported, the next necessary step is in most cases to georeference or to project it, so that all spatial data sets of a project belong to one single coordinate system. Coordinate transformation of *Raphael GCS* is based on two alternative cartographic projection libraries, the GeoTrans library developed by the National Imagery and Mapping Agency, and the Proj.4 library initiated by the U.S. Geological Survey. Both libraries work for raster and vector data and provide various projections for free definable cartographic parameters.

- **Proj.4 - Grid: Coordinate Transformation for Grids.**
- **Proj.4 - Dialog, Grid.**
- **Proj.4 - List of Grids: Coordinate Transformation for Grids.**
- **Proj.4 - Dialog, List of Grids.**
- **Proj.4 - List of Shapes: Coordinate Transformation for Shapes.**
- **Proj.4 - Dialog, List of Shapes.**
- **Proj.4 - Shapes: Coordinate Transformation for Shapes.**
- **Proj.4 - Dialog, Shapes.**

Projection - Tools for the georeferencing of spatial data (grids/shapes)

- **Create Reference Points [Interactive]:** Digitize reference points for georeferencing grids, images and shapes. Click with the mouse on known locations in the map window and add the reference coordinates. After choosing 4 or more points, stop the iterative module execution by unclicking it in the modules menu.
- **Georeferencing - Grids:** Either choose the attribute fields (x/y) with the projected coordinates for the reference points (origin) or supply a additional points layer with correspondend points in the target projection.
- **Georeferencing - Move Grids [Interactive].**
- **Georeferencing - Shapes.**

9. General Terms and Conditions

1. **SELLER.** AVNTK S.C.
2. **ACCEPTANCE.** This quotation constitutes seller's offer to buyer and is expressly limited to the terms and conditions stated herein. Any order received subsequent to this offer shall be an acceptance of the offer. Seller objects to any additional and/or different terms contained in buyer's response. This offer expires sixty (60) days from its date unless products are subsequently shipped by seller and accepted by buyer.
3. **PRICES AND TERMS OF PAYMENT.** The prices for the products and/or services are stated on the face of this contract. Payment shall be made unless otherwise stipulated in cash in full no later than thirty (30) days from the date of shipment. Payment not made when due shall bear interest at the rate of one and one half percent (1.5%) per month from date of shipment until paid in full. If buyer defaults in any payment when due, or in the event any bankruptcy or insolvency proceedings involving buyer are initiated by or against buyer, then the whole contract price shall immediately become due and payable upon demand, or seller at its option without prejudice to its other lawful remedies, may defer delivery or cancel this contract or any unfulfilled orders.
4. **CANCELLATION AND/OR CHANGES.** If buyer returns a product for credit, buyer shall be subject to a twenty percent (20%) restocking charge. If buyer changes the delivery date to exceed six (6) months from the original date, seller may charge the price of the products.
5. **TAXES AND OTHER CHARGES.** Buyer shall pay any taxes imposed by any governmental authority in addition to the prices quoted herein unless buyer gives seller a signed exemption certificate or direct pay permit.
6. **SHIPMENT AND INSURANCE.** Buyer shall be responsible for all loading, freight, shipping, forwarding and handling charges, licenses, import duties and taxes applicable to the delivery of the products. Seller will hold title to the products and bear risk of loss until buyer receives products.
7. **DELIVERY DATE.** The delivery date stated on the face of this contract is an estimate only. Seller will make reasonable efforts to meet the estimated delivery date; however, seller shall not be liable for failure to meet it, unless otherwise stipulated.
8. **FORCE MAJEURE.** Seller shall not be liable for any failure to perform due to causes beyond its reasonable control including but not limited to fire, accident, war, pandemics, labor dispute, shortages, embargo, delayed delivery by suppliers, delay in transportation, inability to secure labor or materials, acts of government whether foreseen or unforeseen. Should any of these events occur, seller may at its option cancel buyer's purchase order with respect to undelivered products, or extend the delivery date for a period equal to the time lost because of the delay.
9. **DESCRIPTIVE LITERATURE.** Specifications are subject to change without notice, unless otherwise stipulated.
10. **FEDERAL ACQUISITION REGULATIONS.** If seller supplies certain products to the U.S. federal government under contract to the General Services Administration, applicable Federal Acquisition Regulations (FAR) apply to such contracts.
11. **WARRANTY AND EXCLUSIONS.** The warranty for products produced by seller and supplied under this contract is limited to the system warranty incorporated by reference, a copy of which is provided in the product documentation and/or upon request. Seller warrants that each product it sells buyer is free from defects in labor and materials and shall conform to its product specifications.

Items marked with an asterisk (*) on the face of the quotation are not produced by seller. Seller is not responsible for the installation, operation or warranty of these products although the manufactures of these products may provide their own warranties to you.

THIS WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER EXPRESS AND IMPLIED WARRANTIES, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND ANY OTHER OBLIGATIONS OR LIABILITIES WHETHER IN CONTRACT, WARRANTY, NEGLIGENCE OR OTHERWISE. SELLER SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, PUNITIVE, CONSEQUENTIAL, OR INCIDENTAL DAMAGES INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS AND BUSINESS INTERRUPTION ARISING OUT OF THE USE OR INABILITY TO USE PRODUCTS OR SERVICES SOLD HEREUNDER.

Seller's liability shall in no event exceed the cost of the products sold under this contract.

12. SOFTWARE LICENSES. Title to licensed products shall remain with seller and shall be subject to software license terms provided with software.
13. AUTHORITY TO EXPORT. Buyer shall not export or re-export technical data, software or products supplied by seller in violation of applicable export regulations. Buyer who exports from the U.S., Mexico or United Kingdom products purchased hereunder assumes all responsibility for obtaining any required export authorization and payment of applicable fees.
14. TECHNICAL INFORMATION. Any sketches, models, samples, designs, algorithms, production processes or techniques submitted or disclosed by seller shall remain the property of seller, and shall be treated as confidential information. No use or disclosure of such information shall be made without the express written consent of seller.
15. GOVERNING LAW AND FORUM. These terms and conditions shall be governed by the laws of the United Kingdom of Great Britain and Northern Ireland, where goods or services originated in full or in part therein. The rights and obligations of the parties shall not be governed by the 1980 U.N. Convention on Contracts for the International Sales of Goods. Buyer consents to venue and jurisdiction over any action related to this contract at the relevant United Kingdom Court, for goods or services originated in full or in part in the United Kingdom of Great Britain and Northern Ireland. For goods or services originated in full from other countries where the seller or associated companies market its products or services, these terms and conditions shall be governed by the laws of the country where this contract is agreed to by both the buyer and seller.
16. MODIFICATION. No addition to or modification of any provision of this contract shall be binding upon seller unless made in writing and signed by a duly authorized representative of seller.
17. SEVERABILITY. Any provision(s) of this contract which in any way contravenes the law of any country in which this contract is effective shall to the extent of such contravention of law, be deemed separable and shall not affect any other provision or the validity of this contract.

Prepared by MFG.
Wednesday, February 10, 2010

Specifications are subject to changes without prior notice.

Third party brands and names are the property of their respective owners.

© Copyright AVNTK S.C., 2010