New image processing tools for GRASS

Markus Neteler*, Daniel Grasso**, Ivan Michelazzi**, Luca Miori**, Stefano Merler*, Cesare Furlanello*

* ITC-irst, Via Sommarive, 13, 38050 Povo (Trento), Italy, tel. ++39-0461-314-520, fax ++39-0461-314-591,
e-mail neteler *at* itc *dot* it
** University of Trento, 38050 Povo (Trento), Italy

Abstract

In this paper we present a suite of new image processing tools for GRASS. New modules are suggested to support improved and semi-automated geocoding of vertical imagery. The ortho-rectification procedures have been extended to rectify oblique imagery from digital hand-held cameras for rendering purposes. Multi- and hyperspectral image analysis has been implemented to derive landuse/landcover maps at subpixel resolution. Image fusion with the Brovey transform is shown. We finally show high performance SMAP image classification on an openMosix cluster.

Keywords: Remote sensing, GIS, image registration, orthophoto, spectral unmixing, image fusion, GRASS on openMosix Cluster

1 Introduction

Remote sensing at various scales plays a major role in spatio-temporal earth surface monitoring. A wide range of satellite platforms as well as film and digital aerial photography provide a huge data repository. Digital image processing in GRASS has been enhanced by a suite of tools as presented in this paper. The integration of remote sensing tools into a full Geographic Information System (GIS) provides a major advance over stand-alone solutions. The new methods are applicable both to aerial and satellite imagery.

2 Image registration

While image registration support has been present in GRASS ([10]) for many years, there have been some restrictions in the flexibility of the provided tools. A basic problem is the time consuming manual identification of ground control points (GCPs). To overcome this problem, two new methods have been implemented which address the GCP identification within time-distant images as well as the semi-automatization of the GCPs search.

2.1 Image registration by homography

Image registration of heterogeneous data sources such as historical images and present-day orthophotos often turns out to be complicated due to differences in survey time, camera type and the evolution of the observed landscape. The search of ground control points becomes very time consuming and may be even impossible for some part of the image scene. The common way of registering images is to find corresponding points (GPCs). An alternative method is the selection of corresponding lines. Linear structures are more easily identifiable by human eye than point structures ([2]). In case of image registration of historical images, substantial changes in the landcover/landuse may render point identification to be impossible while linear structures such as bridges, streets, and railroads still can be selected. A combined method has been implemented in the new modules `i.linespoints` which permits identification of common ground control points in combination with corresponding lines. A subsequent run of the new module `i.homography` allows for homography based image rectification ([3]).

Homography is a simple linear projective transformation of four known points in one planar system (x, y) to another planar reference system (x', y'). Thus, this transformation is applicable only to images covering flat terrain. In the new `i.linespoints`, the transformation, which was originally developed

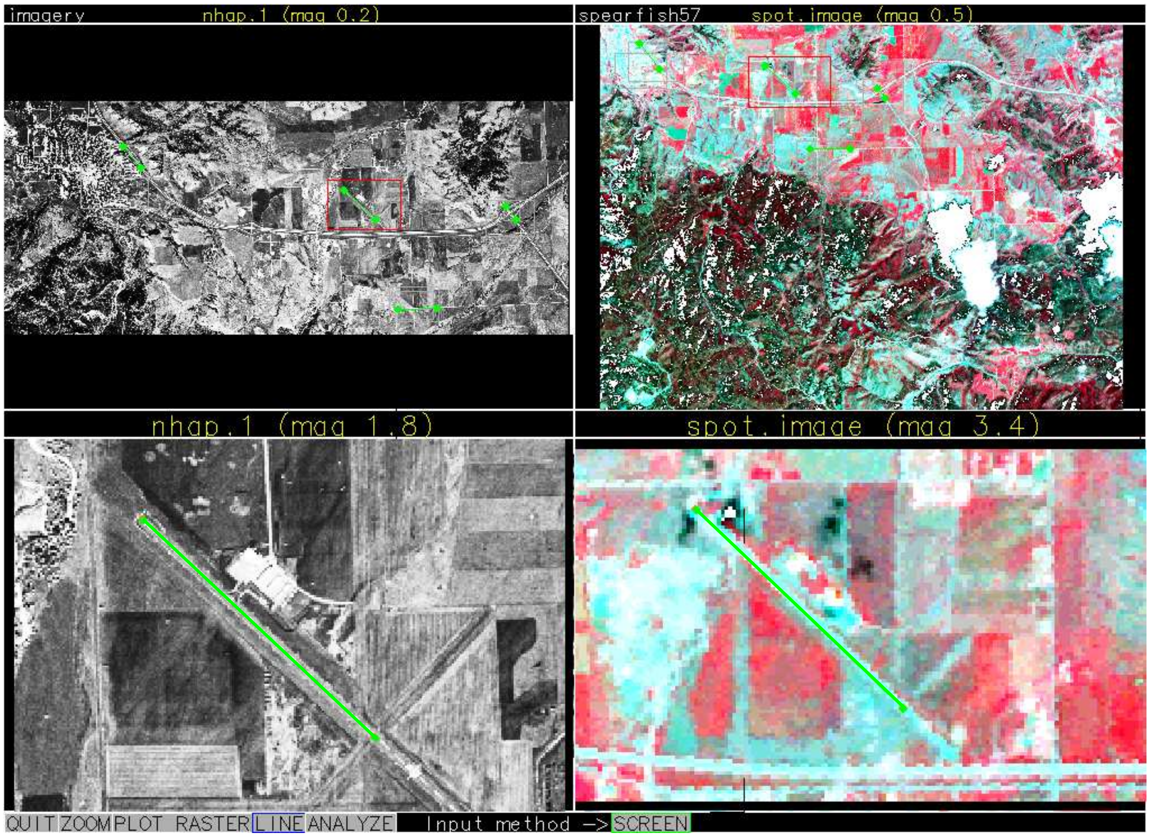


Figure 1: Image registration with homography: Selecting corresponding lines with `i.linespoints` (left: NHAP, right SPOT MSS).

for points only has been extended to homogeneous equations which describe a line (t, u). To unify the two techniques, we developed a method which gives a different weight to points and lines (extending the approach of [8]) according to a c weighting parameter. To minimize the transformation error, the *Brent* minimization algorithm was used to determine the parameter c. Another related method to calculate the mean error was implemented in order to provide this value to the user for both points and lines types. Additionally a test set of independent points can be specified and transformed to verify the overall quality of the rectification for other parts of the image scene.

Figure 1 illustrates the selection of corresponding lines in very heterogeneous images with `i.linespoints`. The subsequent use of `i.homography` rectifies the image to the target LOCATION.

2.2 Semiautomated image registration by Fast Fourier Transform

The classical image registration is done by searching a certain number of corresponding points (ground control points, GCPs) between two images. One image is selected as reference images. These points are then used to define a transformation function, usually a polynomial, to rectify the unregistered image. In current GRASS versions these GCPs have to be identified manually.

The new module `i.coregister` provides an alternative, semiautomated approach to find corresponding points in two overlapping images ([7]). GCPs are found by maximizing the cross-correlation within master-slave search windows between the two images. The cross-correlation is calculated by a frequency analysis of the Fourier transformed search windows.

In order to obtain high registration accuracy, first two regions are roughly indicated on screen by mouse, with very general requirements to image dimensions and overlapping zone characteristics. After determining the size of these search windows (S_d, see Fig. 2), the search window jump distance (S_j, jump in pixels when moving the search window) and the search window border (S_b, minimum distance to keep from border, set to

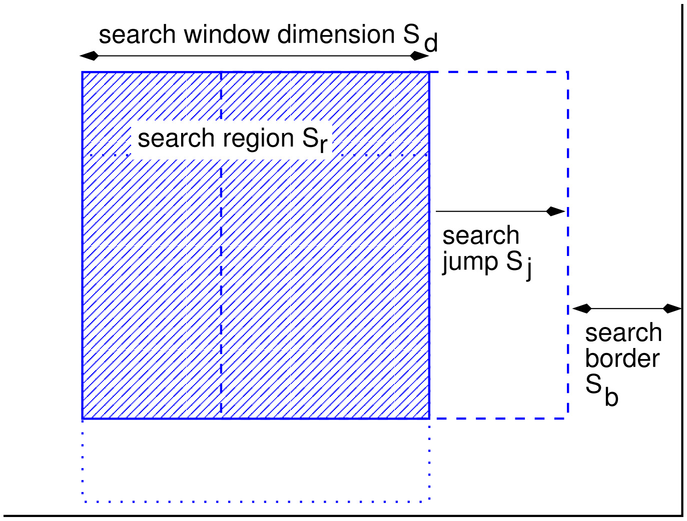


Figure 2: Definition of search windows in FFT based image correlation (`i.coregister`)

$S_d/2$) the algorithm searches for the shift maximizing the cross-correlation between the two search windows. Let $I_1(x,y)$ and $I_2(x,y)$ be the current master and slave search windows respectively. The cross-correlation is defined as

$$H(t_1,t_2) = \int_{S_r} I_1(x,y) I_2(x+t_1, y+t_2)\, d\mu, \tag{1}$$

where S_r is the search region and t_1 and t_2 represent the horizontal and vertical shifts respectively. The convolution theorem allows us to compute the cross-correlation in terms of the Fourier Transforms of I_1 and I_2:

$$H(t_1,t_2) = \mathscr{F}^{-1}(\mathscr{F}(I_1)\mathscr{F}^{\star}(I_2)), \tag{2}$$

where $\mathscr{F}$ indicates the Fourier Transform and $\mathscr{F}^{\star}$ its complex conjugate. The position of the maximum correlation value, $(\bar{t}_1, \bar{t}_2) = \max_{(t_1,t_2)} H(t_1,t_2)$, determines the position of the GCP for the current pair of master/slave search windows.

The set of GCPs is saved into the common GRASS POINTS structure for later use with `i.rectify`. Fig. 3 shows an example for the automated search of GCPs by cross-correlation in a user-defined search region.

The list of GCPs generated by `i.coregister` can optionally be converted into the 3D POINTS structure of `i.ortho.photo` by a new script `i.pointsfile2orthopointsfile.sh` which merges the POINTS file with a user specified camera file and elevations which are automatically queried from a DEM.

2.3 Image registration of oblique aerial photos from digital hand-held cameras

A new application of the `i.ortho.photo` algorithm is proposed for the registration of oblique imagery as produced by handheld digital cameras ([6]). The underlying idea is to improve the visual perception of perspective rendering based on orthophotos. While oblique rendering using a digital elevation model and orthophotos usually suffers from perspective displacements, we show that digital photos taken by cheap digital cameras can be geocoded and used to improve the visual perception.

The existing ortho-rectification algorithm has been tested for

- usage of digital images which are lacking fiducial marks;
- estimation of the focal length of digital hand-held cameras;
- estimation of exterior orientation of oblique photographs taken by a digital hand-held camera.

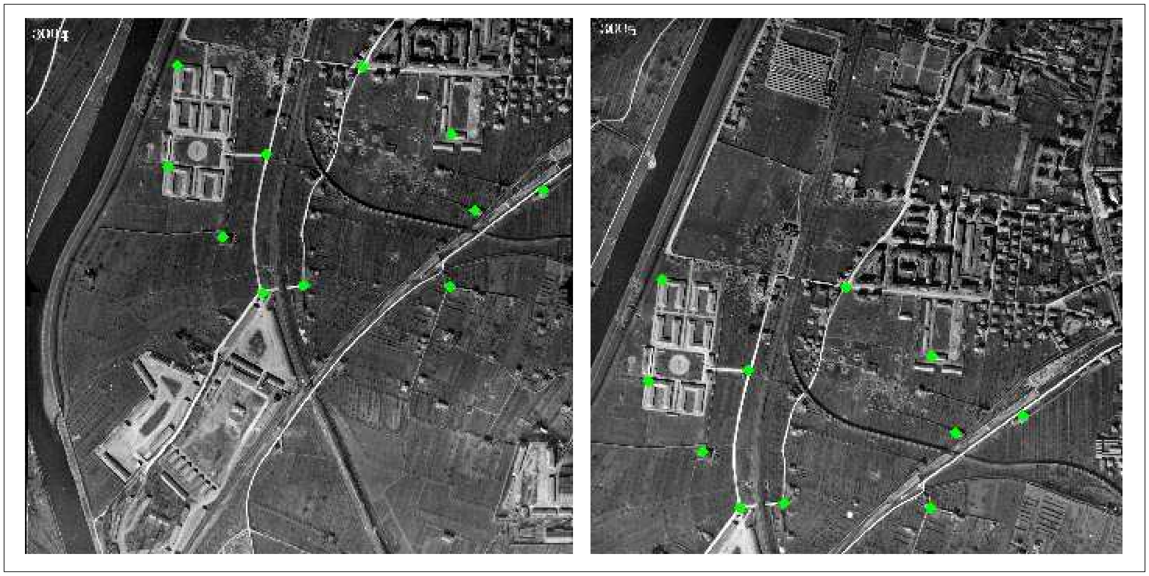


Figure 3: Automated search of GCPs by cross-correlation in a user-defined search region (`i.coregister`)

Most digital hand-held cameras create photos in JPEG format with EXIF (Exchangeable Image File) extension which stores metadata of the camera. Using a *JHEAD EXIF JPEG camera setting parser*[1] camera parameters can be extracted, such as shutter speed, the distance the camera was focused at, focal length and the calculated 35 mm equivalent focal length, image size and resolution, the time and date the picture was taken, as well as the camera maker and model.

The estimation of the parameters of the exterior orientation is done using the "Space Resection by Collinearity" method following the approach of [18]. Space Resection is a method of determining the six elements of exterior orientation of a single tilted photograph. Based on least squares techniques the most probable values for the six elements are determined.

In order to rectify an oblique image with `i.ortho.photo`, the following modifications to the standard procedure have to be applied:

- the focal length is taken from the JHEAD output;
- the number of fiducial marks is set to four, their positions are centered on each side of the photo. To get the position in millimeters, the relation between the longer and the shorter side of the photo is considered (in case of a non-quadratic photo), as well as the resolution versus the photo size in pixels;
- initial parameters for the exterior orientation have to be estimated and entered into the related screen of `i.ortho.photo`:
  - the x, y, z coordinates of the camera position can be easily estimated from DTM with `d.what.rast` or the "What's here" functionality of `nviz`;
  - the orientation angles (ω, ϕ, κ) of the camera can be estimated with `d.measure`;
- the ground control points are identified as usual, however, the user has to take care to select appropriate points in both the oblique photo and the map/orthophoto reference (usually object corners).

Finally the ortho-rectification is performed. A detailed report of the parameter estimation can be obtained by recompilation of `i.ortho.photo` with debug option enabled. Fig. 4 shows an example for the registration of an oblique photo.

[1] JHEAD EXIF JPEG camera setting parser, `http://www.sentex.net/~mwandel/jhead/`

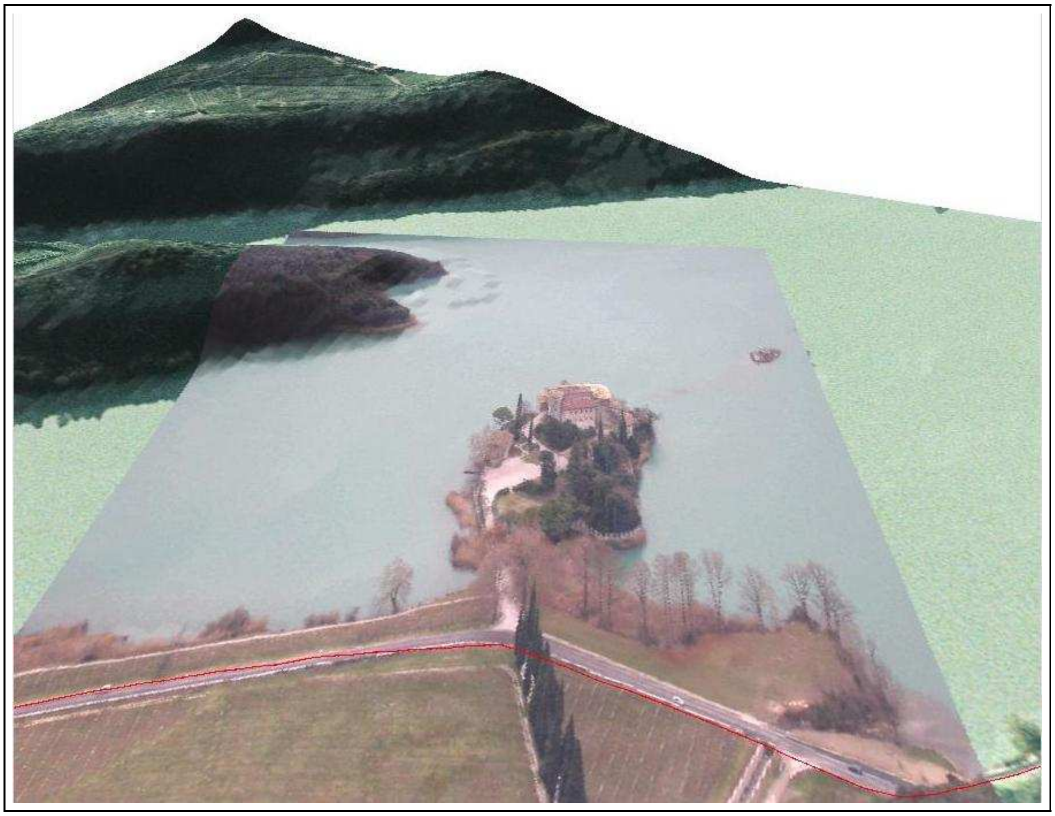


Figure 4: Perspective rendering of an oblique photo taken by a digital hand-held camera (registered by `i.ortho.photo`), the photo is draped over an orthophoto

3 New image classification tools with subpixel accuracy

In this section of the paper we present two methods for analyzing multi- and hyperspectral data. Hyperspectral sensors divide the waveband-range of interest into hundreds of contiguous narrow bands. This spectral sampling is then subject to signature analysis, usually based on a spectral library containing a large set of laboratory spectra or on geometrical search for pure spectra in the image scene itself. The data in each pixel can be understood as a cube of two spatial dimensions and a third spectral dimension, the wavelength. One or many objects are covered by a pixel, resulting in mostly mixed spectral signals of objects within each pixel.

A common model to describe these mixed signals is assumption of linear mixing (linear mixture model, LMM). The methods of spectral angle mapping and spectral unmixing decompose the mixed signal into a set of known spectra (endmembers) from the spectral library. This results in a classifier with subpixel accuracy as for each endmember an abundance map is produced. Difficulties arise from various factors such as dependence on sun angle, atmospheric absorption and scattering, reflections and shadowing, as well as spatial and spectral aberrations in the sensor [14].

3.1 Spectral Angle Mapping

Spectral angle mapping has been implemented in the new module `i.spectral.sam` ([9]). For a set of bands the algorithm is calculating the angles to a set of object spectra which are read from a spectral library. SAM is invariant to unknown multiplicative scalings and, consequently, is invariant to unknown deviations that may arise from different illumination and angle orientation ([5]). Details are explained in the next section, see eq. 12.

3.2 Spectral Unmixing

Spectral unmixing for landuse/landcover mapping at subpixel precision has been implemented in the module `i.spectral.unmix` ([9]). Multi- and hyperspectral data can be analyzed against a spectral library (e.g. USGS Digital Spectral Library[2], ASTER spectral library[3]). The module generates as many abundance maps as object spectra are used. The implemented algorithm assumes the linear mixture of object spectra (also called "endmembers") within each pixel.

The general case is a set of linear equations with m equations and n endmember spectra [15]. The number of equations m is identical to the number of bands of the sensor. The fractions of endmembers represent the percentage of landuse/landcover within the actual pixel. The number of endmember which can be extracted is as high as the number of bands. Written as a general system of equations:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & = & b_m \end{array} \tag{3}$$

The rows correspond to the bands (m bands) and the columns to the endmember spectra (n endmembers). The constants a_{ij} represent the reflection value in the ith band ($0 \le i \le m$) of the jth object ($0 \le j \le n$). The percentage of an endmember fraction within a pixel is x_j while b_i describes the overall pixel value of the ith band. This can be rewritten as:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \tag{4}$$

and simplified to:

$$\mathbf{Ax} = \mathbf{b} \tag{5}$$

Here **A** depicts the matrix of reference spectra, the vector **x** the percentage fraction of spatial distribution within the pixel and the vector **b** the pixel values in the image scene.

To solve this system of equations, it is required to have $m \le n$ which means that the number of objects to be analyzed can be lower or as much as the number of bands available. In case of $m = n$ the matrix **A** is quadratic. An additional condition is that the equations should be linearly independent, i.e the matrix **A** needs to have a large rank.

The unknown vector **x** can be calculated as:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{6}$$

But to get physically meaningful results, we must enforce two boundary conditions:

First, the sum of all fractions must be 100% (full additivity, assuming the case that at least one reference spectrum is found in each pixel):

$$\sum_{j=1}^{n} x_j = 1 \tag{7}$$

Second, the fractions must be zero or positive as no negative area sizes exist in nature (non-negativity):

$$x_j \ge 0 \quad \text{for all } j. \tag{8}$$

The ideal solution of the rewritten eq. 6 is zero:

$$\mathbf{A}^{-1}\mathbf{b} - \mathbf{x} \overset{!}{=} 0 \tag{9}$$

[2] USGS Digital Spectral Library, `http://speclab.cr.usgs.gov/spectral-lib.html`

[3] ASTER spectral library, `http://speclib.jpl.nasa.gov`

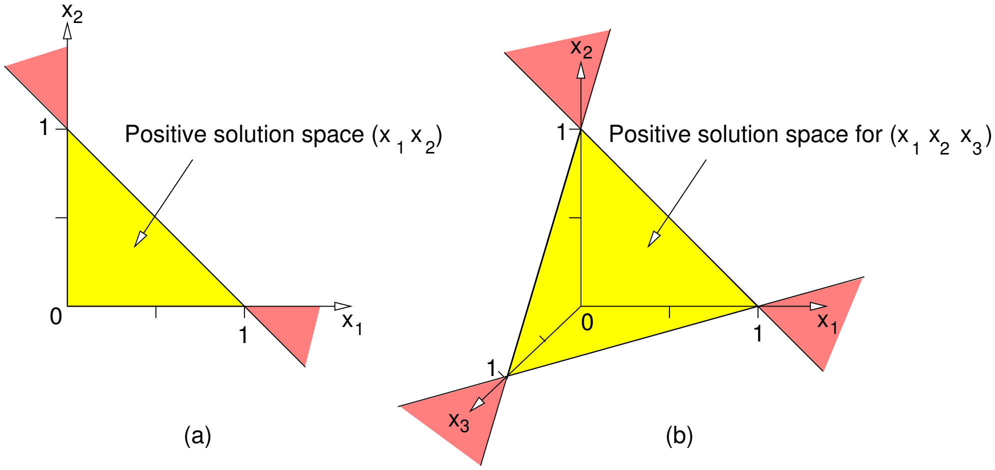

x2
1
0
1
x1
Positive solution space (x 1 x 2)
(a)
x2
1
0
1
x1
1
x3
Positive solution space for (x 1 x 2 x 3)
(b)

Figure 5: Permitted (bright) and undesired (dark) negative solution space of a two-dimensional (a) and of a three-dimensional spectral unmixing (b) with constraints (after [16])

But due to limitations in the accuracy of the reference spectra, limitations in the assumption of linear mixture and natural deviations of the object spectra from the reference spectra, the eq. 6 cannot be solved. To overcome this problem, an error term ε is introduced to cover the unidentifiable object spectra and noise:

$$\mathbf{A}^{-1}\mathbf{b} - \mathbf{x} = \varepsilon \tag{10}$$

Because vector $\mathbf{x}$ and error ε are unknown, the equation can be solved by minimizing the error ε through least-squares. The error function $\mathbf{F}$ to be minimized can be written as:

$$\mathbf{F} = \sum_{i=1}^{m} \varepsilon_i^2 \tag{11}$$

With minimization of the error function $\mathbf{F}$, the vector $\mathbf{x}$ which represents the percentage fraction of the reference spectra in the actual pixel can be calculated from eq. 5. The methods have been implemented to calculate the area fractions x_j while minimizing the error and respecting the two boundary conditions as given in eq. 7 and eq. 8.

It is recommended to test for linear dependencies beforehand the matrix of reference spectra $\mathbf{A}$ using Spectral Angle Mapping (SAM). In this case the selected spectra should be exchanged from the matrix to improve the spectral separability. This is obtained by calculating the angle φ between column vector a_j and a_{j+1} in the matrix, i.e. between individual reference spectra ([16]):

$$\varphi = cos^{-1}\left(\frac{\mathbf{a}_j^T \mathbf{a}_{j+1}}{||\mathbf{a}_j||_2 \cdot ||\mathbf{a}_{j+1}||_2}\right) \tag{12}$$

The length of the vectors represents the signal power, i.e. the reflectivity of the observed object surface. If the spectra are too close, i.e. the angle φ is too small, a spectral separation cannot be performed. In this case the vectors are linearly dependent. Less noise and a higher number of bands minimizes the problem.

A mathematical problem arises from the inversion of matrix $\mathbf{A}$ (eq. 6 and eq. 9). Due to limits in the sensor signal and other noise only ill conditioned matrices are expected; classical matrix inversion methods such as Gauss elimination or LR decomposition will fail here. The Singular value decomposition (SVD) proposed by some authors is also inappropriate as it does not fulfill the second boundary condition (eq. 8).

While enforcing the first (sum of areas be 100%) and the second (all area percentages within the positive range, compare eq. 5) boundary conditions the implemented algorithm iteratively minimizes the error ε (*constraint least square method*, [15]). Mathematically the solution of the equation is projected from the undesired negative solution space to the boundary of the desired positive solution space ([16], [4]:8).

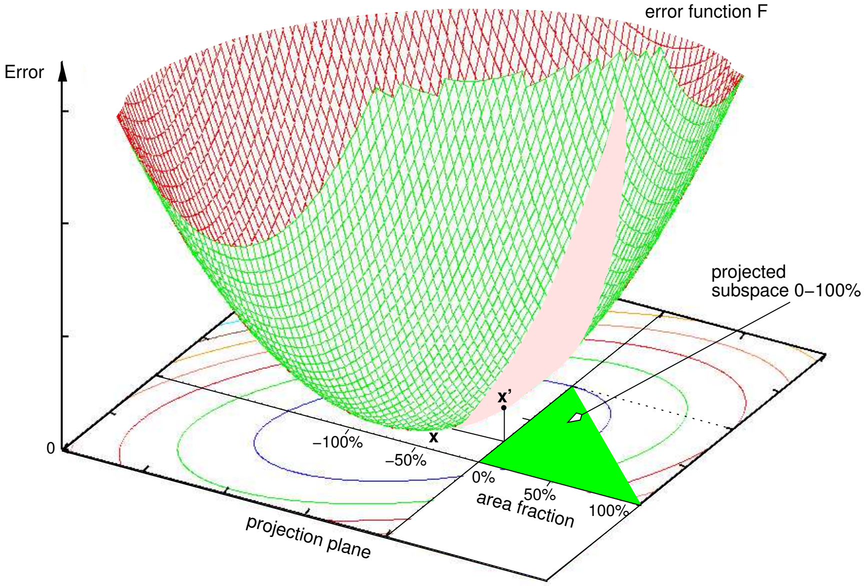


Figure 6: Example for error function **F**, as calculated for a pixel in spectral unmixing of two bands. The figures depicts the case of a numerically correct solution for fraction **x** in the negative range. However, this minimum solution is outside the boundary conditions. The physically desired solution $\mathbf{x}'$ lays within the range of 0% to 100% (triangle on error surface) and is found by projecting into the positive range while keeping $\mathbf{x}'$ as close as possible to error minimum at **x**.

To solve the set of equations the gradient of the error function **F** (eq. 11) is considered. In a multidimensional feature space the function **F** is a hyper-parabola. A three-dimensional case for two spectral bands is shown in fig. 6. The error surface for this example is calculated as:

$$\mathbf{F} = ||(\mathbf{A}\mathbf{x} - \mathbf{b})||_2^2 = (\mathbf{A}\mathbf{x} - \mathbf{b})^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \tag{13}$$

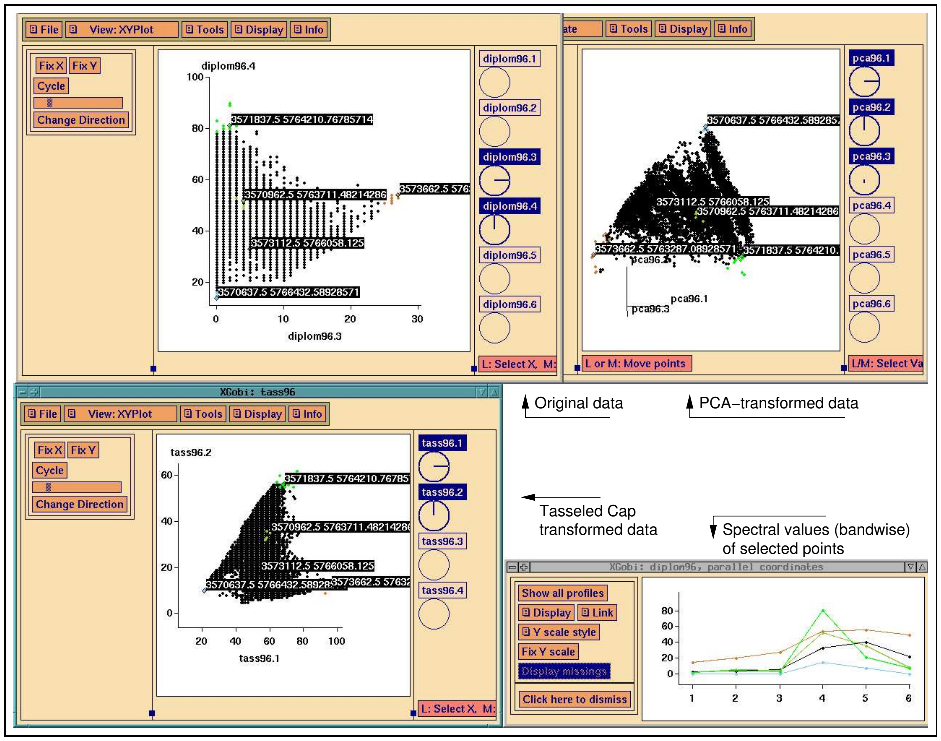
The numerically optimal solution **x** would be the minimum of the error function but it violates the boundary conditions. In the proposed algorithm, a vector $\mathbf{x}'$ is initialized to a seed value. The set of seed values initially define an equivalent area fraction for each endmember and are put into eq. 9. Using the approach of *deepest decent* the calculated error can be minimized. To do so, the values of the vector $\mathbf{x}'$ are lowered by the small value μ in the direction of decreasing slope of the error function. The gradient at position k on the error surface **F** is calculated as (Harteneck 1998, pers. comm.):

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = 2\mathbf{A}^T(\mathbf{A}\mathbf{x}'(k) - \mathbf{b}) \tag{14}$$

The new vector $\mathbf{x}'$(k+1) is calculated for an increment μ as:

$$\mathbf{x}'(k+1) = \mathbf{x}'(k) - \mu \cdot 2\mathbf{A}^T(\mathbf{A}x(k) - \mathbf{b}) \tag{15}$$

In the next iteration the new vector $\mathbf{x}'$ will be inserted again into eq. 9, which determines a new error value. This will lead to a corrected vector $\mathbf{x}'$. We finish the loop when we leave the positive solution space or when we reach the minimum of the error function. The iteration is performed until no further significant change appears in the error minimization. In fig. 6, the search for the optimal solution can be seen as moving towards of the minimum. To spectrally unmix an image scene, this method must be applied to all pixels.



Original data

PCA–transformed data

Tasseled Cap transformed data

Spectral values (bandwise) of selected points

Figure 7: XGOBI multi-session with interprocess communication enabled. The user can select endmembers for spectral unmixing in the extremities of the data clouds (here PCA and Tasseled Cap transform) and analyse the spectra in the parallel plot window

The signature analysis is usually based on a spectral library containing a large set of laboratory spectra or done with a geometrical search for "pure" spectra in the image scene. A geometrical search can be done by various methods, such as using Principal Component Analysis (PCA). Endmembers ("pure" spectra) are located in the extremities of the PCA-transformed multidimensional pixel cloud. They can be extracted interactively with XGOBI software ([17]). This software supports interprocess communication (IPC) which enables the user to run several program sessions in parallel while the sessions maintain communication. The user can interactively query transformed and untransformed data in either session for the identification of endmembers, pixel map coordinates are indicated in all XGOBI windows (see fig. 7). From these coordinates pure pixels can be easily extracted and used as endmembers.

From the created endmember file (either from a spectral library or from geometrical search within the image scene) the spectral unmixing is performed. The result is a set of endmember abundance maps describing pixelwise the area percentage of each endmember. The algorithm is implemented in `i.spectral.unmix`.

4 Image fusion with Brovey transform

In remote sensing systems parts of the spectrum are observed in different portions to minimize the impact of noise. While multispectral sensors provide high spectral resolution at lower spatial resolution, panchromatic sensors collect data at high spatial resolution and lower spectral resolution. Numerous algorithms have been proposed to combine these data with best characteristics of both sensor types. One of these algorithms, the Brovey transform, has been selected for script based implementation in GRASS.

The Brovey transform is a color transform which normalizes three input bands and multiplies the result with a higher resolution band (panchromatic) ([12; 11]):

$$\mathbf{D}N_{fused} = \frac{DN_{b1}}{DN_{b1} + DN_{b2} + DN_{b3}} * DN_{pan} \qquad (16)$$

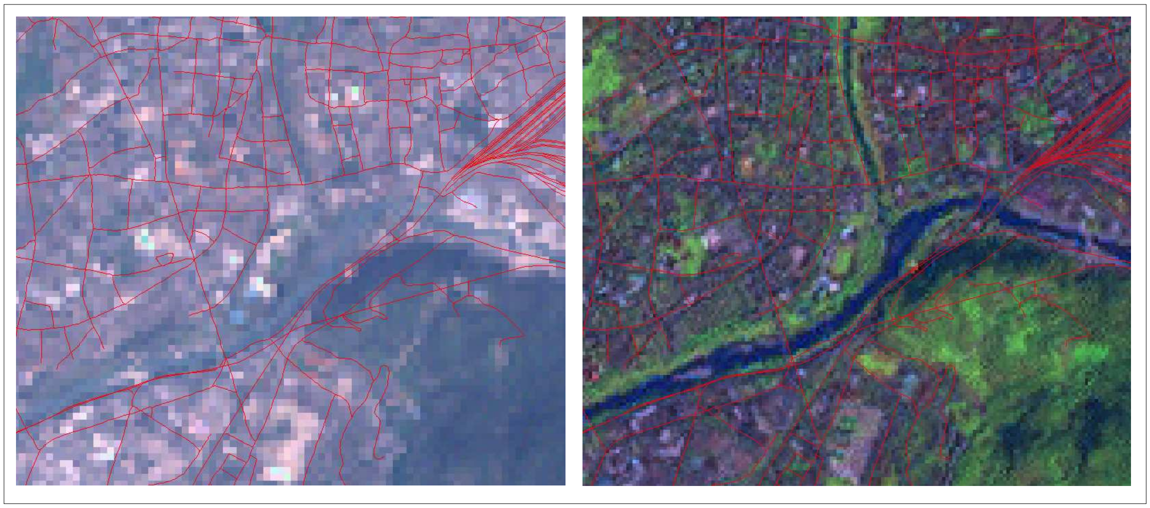


Figure 8: Left: LANDSAT-7 RGB composite at 28.5m; right: LANDSAT-7 RGB/PAN Brovey fusion at 14.25m

The Brovey algorithm holds an intermediate position between band arithmetic and component substitution methods. A new script `i.fusion.brovey` has been integrated to GRASS 5.7 to support PAN sharpening of multispectral satellite such as that from LANDSAT-7, Quickbird and SPOT. The algorithm performs image fusion of the high resolution panchromatic channel with the multispectral channels at lower resolution. The resulting pseudo-RGB channels are saved at the high resolution of the panchromatic channel and have a near-natural color table. An example of LANDSAT-7 fusion is shown in fig. 8.

5 High performance image classification on openMosix cluster

A high performance solution for image classification with GRASS at meso-scale but with high spatial resolution was developed. A script-based approach to run the standard, unmodified GRASS on an GNU/Linux based openMosix cluster (20 PCs, 40 CPUs) has been implemented to classify multispectral color orthophotos with SMAP algorithm (*Sequential maximum a posteriori estimation*, [1], [13]:440).

openMosix (`www.openmosix.org`, published under GPL) is a Linux kernel extension for single-system image clustering. Once openMosix is installed and the cluster nodes are connected by a local network, the nodes in the cluster communicate to optimally distribute the workload. Processes originating from any node can migrate to any other node if the current node is too busy compared to others. openMosix continuously attempts to optimize the resource allocation. New nodes can be added while the cluster is running and the cluster will automatically begin to use the new resources. Due to the distributed computing concept of openMosix, the cluster behaves much like a symmetric multi-processor (SMP).

In order to run GRASS in parallel without code modifications, a multiple sequential approach was used. Within the job launch script the GRASS environment variables are defined. It is important to run each job in its own mapset to avoid one job influencing the settings of another job. The definition section is followed by the GRASS analysis (in our case the SMAP classification). To finish, the launch script removes the temporary files and copies the resulting classification map to the PERMANENT mapset. It uses UNIX commands as the original `g.copy` command may fail due to race conditions (here: concurrent writing of files) with other jobs copying results to the PERMANENT mapset at the same time.

The approach was tested in a large study area, the Autonomous Province of Trento (Italy), which is of approximately 6200 km^2 area. This area is covered by 280 RGB orthophotos at a resolution of one meter per pixel. In tests, the required time to analyze these orthophotos at the given resolution was reduced from estimated 110 days on a single CPU to 5 days on the ITC-irst openMosix cluster.

Acknowledgments

Prof. Dr. Bruzzone at University of Trento supervised the work developed in Section 2.1 and Section 2.2. S. Fontanari and S. Menegon at ITC-irst gave technical assistance in the related source code implementations. Markus Neteler was supported by the FUR-PAT Project WebFAQ.

References

[1] C. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, 1994.

[2] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey. *Graphical methods for data analysis*. Chapman & Hall, 1983.

[3] D. Grasso. Registrazione di immagini telerilevate mediante strutture lineari e ground control points. Master's thesis, University of Trento, Italy, 2004.

[4] S.E. Hobbs. Linear mixture modelling solution methods for satellite remote sensing. Technical Report 9603, COA, Bedford, 3 1996.

[5] N. Keshava and J.F. Mustard. Spectral Unmixing. *IEEE Signal Processing*, 19(1):44–57, 2002.

[6] I. Michelazzi. Geocodifica ed elaborazione di immagini oblique per visualizzazione 3D in ambiente GIS. Master's thesis, University of Trento, Italy, 2004.

[7] L. Miori. Tecniche automatiche per la co-registrazione di immagini telerilevate. Master's thesis, University of Trento, Italy, 2004.

[8] V. Murino, U. Castellani, A. Etrari, and A Fusiello. Registration of very time-distant aerial images. *Proc. of the IEEE Int. Conf. on Image Processing (ICIP 2002, Rochester, NY, Sept. 22-25)*, III:989–992, 2002.

[9] M. Neteler. Spectral Mixture Analysis von Satellitendaten zur Bestimmung von Bodenbedeckungsgraden im Hinblick auf die Erosionsmodellierung. Master's thesis, University of Hannover, Germany, 1999.

[10] M. Neteler and H. Mitasova. *Open Source GIS: A GRASS GIS Approach*. Number 773 in SECS. Kluwer Academic Publishers, Boston, 2nd edition, June 2004.

[11] C. Pohl and J.L. van Genderen. Multisensor image fusion in remote sensing: concepts, methods and application. *Int. J. of Rem. Sens.*, 19:823–854, 1998.

[12] N.E.G. Roller and S. Cox. Comparison of Landsat MSS and merged MSS/RBV data for analysis of natural vegetation. *Proc. of the 14th International Symposium on Remote Sensing of Environment*, pages 1001–1007, 1980.

[13] R. Schowengerdt. *Remote sensing: Models and methods for image processing*. Academic Press, 2 edition, 1997.

[14] G. Shaw and D. Manolakis. Signal processing for hyperspectral image exploitation. *IEEE Signal Processing*, 19(1):12–16, 2002.

[15] Y.E. Shimabukuro and J.A. Adams. The least-squares mixing models to generate fraction images derived from remote sensing multispectral data. *IEEE Transactions on Geoscience and Remote sensing*, 29(1):16–20, 1991.

[16] Y. Sohn and R.M. McCoy. Mapping desert shrub rangeland using spectral unmixing and modeling spectral mixtures with TM data. *Photogrammetric Engineering & Remote Sensing*, 63(6):707–716, 1997.

[17] D.F. Swayne, D. Cook, and A. Buja. XGobi: Interactive dynamic data visualization in the X Window System. *Journal of Computational and Graphical Statistics*, 7(1):47–67, 1998.

[18] P. Wolf. *Elements of Photogrammetry With Air Photo Interpretation and Remote Sensing*. McGraw-Hill, New York, 2nd edition, 1983.