University of Rome "La Sapienza"
Faculty of Engineering

DITS - Department of Hydraulics, Transportations and Roads
Geodesy and Geomatic Area

PhD Course in Transportations and Infrastructures

PhD Thesis

Orientation models of optical High Resolution Satellite Imagery: definition, implementation and validation of original algorithms

PhD Supervisor
Prof. Mattia Giovanni Crespi

PhD Student
Francesca Fratarcangeli

Rome, 12 Marzo 2010

“A Papà e Mamma”
“A Emiliano”

Contents

List of Figures

List of Tables

Abstract

High resolution satellite imagery became available to civil users in 1999 with the launch of Ikonos, the first civil satellite offering a spatial resolution of 1 m. Since then other high resolution satellites have been launched, among which there are EROS-A (1.8 m), QuickBird (0.61 m), Orbview-3 (1 m), EROS-B (0.7 m), Worldview-1 (0.5 m) and GeoEye-1 (0.41 m), with many others being planned to launch in the near future.

High resolution satellite imagery is now available in different formats and processing levels at an affordable price, so that they already represent a possible alternative to aerial imagery, for cartographic applications and orthophoto production, especially for areas where the organization of photogrammetric surveying may be critical.

Moreover, an increasing demand for terrain modelling exists so that almost all the satellites have along-track stereo acquisition capability. Many new satellites dedicated to stereo viewing, for example Cartosat-1 (2.5 m), have been launched. This enables the generation of Digital Elevation Models (DEMs) and Digital Surface Models (DSMs), and also for 3D feature extraction (e.g. for city modelling).

The geomatic utilizations of satellite imagery for cartographic applications and terrain modelling require a high level geometric correction through image orientation.

Some fundamental features related to the sensor models and their parameters estimation, both for single images and stereopairs orientation, were addressed and some real applications were discussed.

In details, they were concerned both physical sensor models (also called rigorous models) and generalized sensor models (also called RPC models) for the orientation of basic images (level 1A) and of the image projected onto a specific object surface (usually an expanded ellipsoid derived from the WGS84) (level 1B).

As regards the rigorous models, a thorough investigation of the fundamentals of their functional model was developed and the problem of parameters estimability was concerned, proposing a solution based on SVD and QR decomposition. RPC models were discussed not only with respect possible refinements by zero and first order transformations but also (and mainly) with respect the RPCs generation, based on previously established rigorous model; thanks to SVD and QR decomposition, it was showed that many RPCs are not estimable parameters, therefore they are not necessary to obtain the best achievable accuracy level.

Real applications demonstrated that rigorous and RPC models both for Level 1A and Level 1B imagery can provide orientation accuracy at 1-1.5 pixel level in the horizontal components, and at 1-2 pixel level in the height for stereopairs (even better with Cartosat-1 and slightly worse with EROS-1).

Introduction

A few years ago high resolution satellite imagery became available to a limited number of government and defense agencies that managed such imagery with highly sophisticated software and hardware tools. Such images became available to civil users in 1999 with the launch of Ikonos, the first civil satellite offering a spatial resolution of 1 m. Since then other high resolution satellites have been launched, among which there are EROS-A (1.8 m), QuickBird (0.61 m), Orbview-3 (1 m), EROS-B (0.7 m), Worldview-1 (0.5 m) and GeoEye-1 (0.41 m), with many others being planned to launch in the near future. High resolution satellite imagery is now available in different formats and processing levels at an affordable price. The diverse types of sensors and their growing availability are revolutionizing the role of satellite imagery in a number of applications, ranging from intelligency to insurance, media, marketing, agriculture, utilities, urban planning, forestry, environmental monitoring, transportation, real estate etc. As a possible alternative to aerial imagery, high resolution satellite imagery has also impact in cartographic applications, such as in orthophoto production, especially for areas where the organization of photogrammetric surveying may be critical.

Moreover, an increasing demand for large scale mapping and terrain modelling exists so that almost all the satellites have along-track stereo acquisition capability. Many new satellites dedicated to stereo viewing, for example Cartosat-1 (2.5 m), have been launched. This compensates the limited capacity of very high resolution satellites for three-dimensional point determination and enables the generation of Digital Elevation Models (DEMs) and Digital Surface Models (DSMs), and also for 3D feature extraction (e.g. for city modelling).

However, the possibility of using high resolution satellite images for cartography depends on several factors: mapping specifications, sensor characteristics (geometric and radiometric resolution and quality), types of products made available by the companies managing the satellites, quality of the software used in the

final processing to produce the cartographic products, and quality of the final results.

One of the primary barriers to a wider adaptation and utilization of satellite imagery is the sensor model being able to provide a high level geometric correction through image orientation. The distortion sources can be related to two general categories: the acquisition system, which includes the platform orientation and movement, the imaging sensor optical-geometric characteristics, and the atmosphere refraction, causing a remarkable deviation from the collinearity hypothesis [2].

Two different types of orientation models are usually adopted: the physical sensor models (also called rigorous models) and the generalized sensor models. In the first ones, based on a standard photogrammetric approach where the image and the ground coordinates are linked through the collinearity equations and the involved parameters have a physical meaning. Besides, they require knowledge on the specific satellite and orbit characteristics. On the contrary, the generalized models are usually based on the Rational Polynomial Functions (RPFs), which link image and terrain coordinates by the Rational Polynomial Coefficients (RPCs) and do not need the knowledge about the sensor and acquisition features. The RPCs can be calculated by the final users via a Least Squares (LS) estimation directly from Ground Control Points (GCPs), or proprietarily generated by the sensor managing companies based on their own physical sensor models and dis-tributed to users through imagery metadata. Nevertheless, the first strategy (also called terrain-dependent) is not recommended if a reliable and accurate orientation is required. In the second strategy, they can be generated according to a terrain-independent scenario, using known physical sensor. In order to avoid instability due to high correlations among the coefficients, two different methods can be used: Tichonov regularization or an innovative method based on Singular Value Decomposition (SVD) and QR decomposition, estimating only the strictly required coefficients.

This thesis will discuss many features of the sensor models, both for single images and stereopairs. Specifically, discussions will be focused on the rigorous model for the orientation of the basic image (level 1A) and of the image projected to a specific object surface (usually an expanded ellipsoid derived from the WGS84) (level 1B), both extended to stereopairs in chapter 2. The RPC model for the orientation of single images and stereopairs is discussed in chapter 3. Chapter 4 addresses the tecniques for resolution of LS, while chapter 6 presents application examples of different sensor modelling.

Chapter 1

Rigorous models

The rigorous model is based on a standard photogrammetric approach, i.e., the collinearity equations describing the physical-geometrical image acquisition. It must consider that an image from a pushbroom sensor is formed by many (from thousands to tens of thousands) individual lines, each acquired with proper position (projection center) and attitude values. All the acquisition positions are related by the orbital dynamics. Therefore, the rigorous model is based on the reconstruction of the orbital segment during image acquisition through the knowledge of the acquisition mode, sensor parameters, satellite position and attitude parameters. The approximate values of these parameters can be computed by using the information contained in the image metadata file, delivered with each image. These approximate values are then corrected by a LS estimation process based on a suitable number of GCPs. Moreover, in order to relate the images to the ground coordinates, expressed in an Earth Centered - Earth Fixed (ECEF) reference frame, a translation and a set of rotation matrices depending on orbital parameters (Keplerian elements) and sensor attitude have to be used. These matrices include those needed to shift between sensor, platform, orbital and Earth Centered Inertial (ECI) coordinate systems, while the transformation between ECI and ECEF coordinate systems must take into account precession, nutation, polar motion and Earth rotation matrices [3].

1.1 Coordinate systems

In order to introduce the collinearity equations, the definitions of some coordinate systems are needed [4]:

Sensor system (S): the origin is in the perspective center (whose orbital motion may be describe as if it was the satellite centre of mass), the z-axis is directed from the perspective center to pixel array, the x-axis is approximately tangent to the orbit directed as the satellite motion (see Sections 1.4, 1.5), and y-axis forms a right-handed cartesian system; note that y-axis is approximately parallel to pixel array, a part for the same possible rotation around the perspective center (Fig.1.1a); the principal point is the orthogonal projection of the perspective center onto the pixel array.

Image system (I): is a 2-dimensional system describing a pixel position in an image. The origin is in the upper left corner, and the pixel position is defined by its row (J) and column (I). The column numbers increases toward the right and row numbers increases downwards (Fig.1.1b).

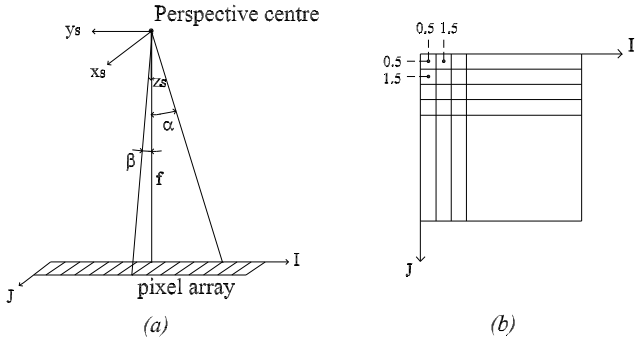


Figure 1.1: Coordinates in the Sensor system (a) and in the Image system (b)

Body system (B): it is aligned to the Flight system (see below) when the angle Roll (φ), Pitch (ϑ) and Yaw (ψ) are zero.

Flight system (F): the origin is in the perspective center, the X-axis is tangent to the orbit along the satellite motion, the Z-axis is in the orbital plane directed towards the Earth center of mass and the Y-axis completes the right-handed coordinate system (Fig.1.2).

Orbital system (O): the $Xo - Yo$ plane coincides to the orbital plane, which is defined by right ascension of the ascending node (Ω) and by the orbit inclination (i). X-axis is along the nodal line, Y-axis and Z-axis complete the

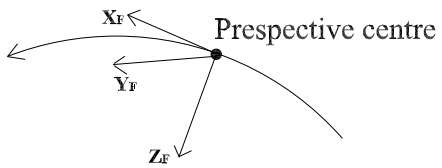


Figure 1.2: Coordinates in the Flight system

right-handed coordinate system. With the hypothesis of Keplerian orbit, Z-coordinate of satellite in the orbital system is zero (Fig.1.3).

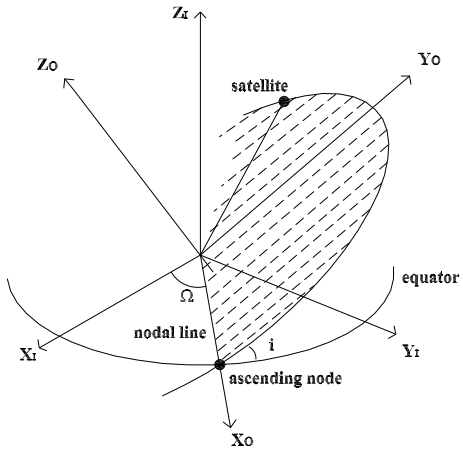


Figure 1.3: Coordinates in the Orbital system

Earth Centered Inertial system - ECI (I): the origin is in the Earth center of mass, the X-axis points to vernal equinox (epoch J2000 - 1 January 2000, hours 12 UT), the Z-axis points to celestial north pole (epoch J2000) and the Y-axis completes the right-handed coordinate system [5], [6], (Fig.1.4*a*).

Earth-Centered Earth-Fixed system - ECEF (E): the origin is in the Earth center of mass, the X-axis is the intersection of equatorial plane and the plane of reference meridian (epoch 1984.0), the Z-axis is the mean rotational axis (epoch 1984.0) and the Y-axis completes the right-handed coordinate system [5], [6], (Fig.1.4*b*).

Geodetic Local system (L): the origin is a chosen point on the ellipsoid

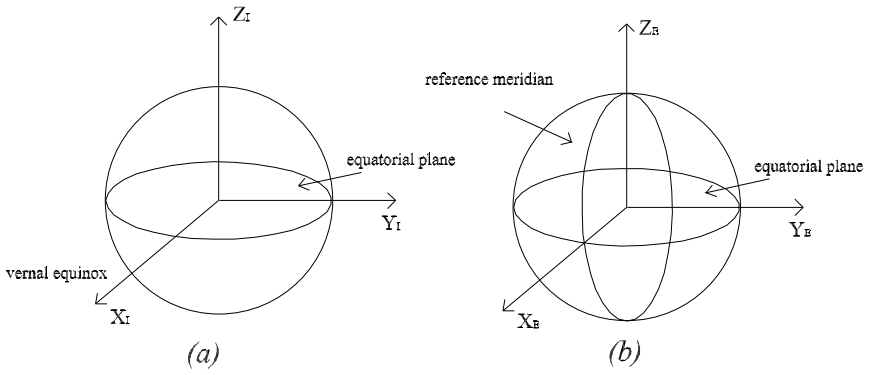


Figure 1.4: Coordinates in the Earth Centered Inertial - ECI (a) and in the Earth-Centered Earth-Fixed - ECEF (b)

(here the WGS84 is used) the N-axis is tangent to the local meridian, E-axis is tangent to the local parallel and h-axis (elevation axis) is along the ellipsoid normal (Fig.1.5).

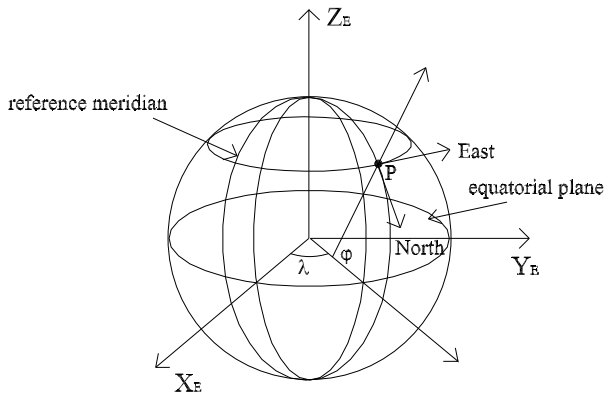


Figure 1.5: Coordinates in the Geodetic Local system

1.2 Orbital parameters

The satellite orbit can be described using the well-known Keplerian elements (Fig.1.6). According to Keplerian laws, a satellite (considered as a material point), under the effect of a gravitational field generated by a mass concentrated in a point, moves in a plane describing an elliptic orbit.

The satellite position at each generic epoch T can be represented by seven parameters [3], [4]:

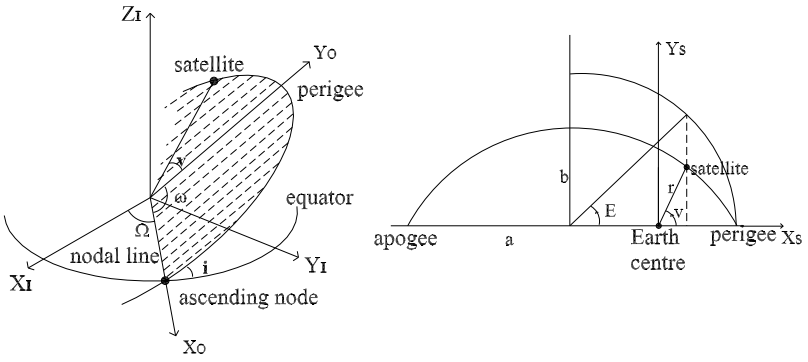


Figure 1.6: Illustration of the Keplerian parameters

semi-major axis (a): it is semi-major axis of the orbital ellipse

orbit inclination (i): it is the angle (positive if counterclockwise) between the orbital plane and the equatorial plane; by convention the inclination is a number between 0 and π

right ascension of the ascending node (Ω): it is the angle (positive if counterclockwise observed from the North Pole) at the center of the Earth from the vernal equinox to the ascending node

eccentricity (e): it is the eccentricity of the orbital ellipse

true anomaly (v): it is the angle measured in the center of the ellipse between the perigee and the position of the satellite at generic epoch T defined to be 0 at perigee

argument of the perigee (ω): it is the angle between the nodal line (intersection between the orbital plane and the equatorial plane) and the semi-major axis, measured in the orbital plane from the ascending node to the perigee

time of the perigee passage (T_p): it is the time referring to the epoch when the satellite is nearest to the Earth

The approximate values of these parameters can be computed thanks to the ephemeris information contained in the metadata file released together with the image.

In some cases (e.g. Ikonos, QuickBird Standard OrthoReady, Cartosat-1 and GeoEye-1) the metadata file, released by the sensor managing Companies does not include the ephemeris file (spacecraft position and velocity every few seconds) but only one satellite mean position which is described by two angles:

azimuth (α): it specifies the satellite position relative to the area that is collected on Earth, it is measured clock wise from the North (Fig.1.7)

elevation (e): it is the angle from the horizon up to the satellite (Fig.1.7) Nevertheless, some main features of the orbit (a, i) are always known, so that

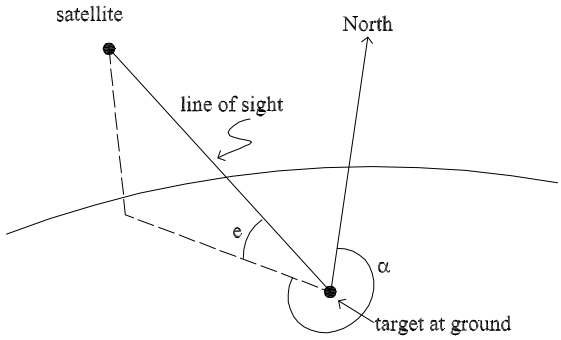


Figure 1.7: Azimuth and elevations angles

the approximate values of the other Keplerian elements may still need to be computed.

1.3 Attitude angles

To define the sensor during the acquisition it is necessary to know its attitude described by the Roll (φ), Pitch (ϑ) and Yaw (ψ) angles, respectively referred as X_F, Y_F, Z_F axes of the Flight system.

The approximate values of these angles are calculated with the metadata file information. The corrections to these approximate values are supposed to be modelled by second order polynomials. Although there is not any physical

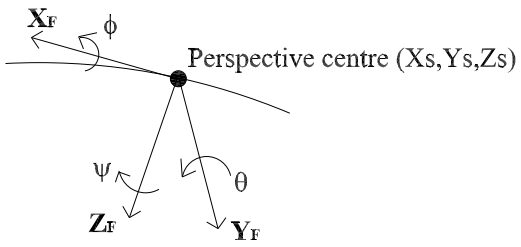


Figure 1.8: The sensor attitude angles (Roll (φ), Pitch (ϑ) and Yaw (ψ))) respectively referred to the axes of the Flight system (X_F, Y_F, Z_F); these attitute angles define the position of the (X_B, Y_B, Z_B) axes of the Body system

meaning in doing this, good results seem to support this choice [4].

$$\begin{cases} \varphi = \tilde{\varphi} + a_0 + a_1\tau + a_2\tau^2 \\ \vartheta = \tilde{\vartheta} + b_0 + b_1\tau + b_2\tau^2 \\ \psi = \tilde{\psi} + c_0 + c_1\tau + c_2\tau^2 \end{cases} \tag{1.1}$$

τ is the time, in seconds, such as $\tau = J_s \cdot \Delta t$ where Δt is the time needed to scan a row on the ground and J_s is the row of the pixel. The nine coefficients (a_i, b_i, c_i) are unknown and need to be estimated with the LS adjustment.

In some case (e.g. Ikonos) the satellite can acquire images at a stated scan azimuth (the exact angle from the starting point of the scan and it is measured clock wise from North) and (e.g. Ikonos, WorldView-1) following a stated scan direction ("forward" or "reverse"); a reverse scan is generally from North to South direction, whereas a forward scan being from South to North (Fig.1.9).

If the scan direction is "forward", $\tau = J_s \cdot \Delta t$, instead if the scan direction is "reverse" $\tau = (n°raw - J_s) \cdot \Delta t$, where $n°raw$ is the number of raws of the image.

If the scan azimuth is 180° and the scan direction is "reverse", the image is collected from North to South, or if scan azimuth is 90° and the scan direction is "forward", the image is collected from West to East. In the first case, the J-axis of image system is directed to the scanning direction, whereas in the latter the scanning direction is perpendicular to the J-axis of the image system (Fig.1.10). So the position of a generic point $p(I, J)$ has to be projected on scanning direction (J_s-axis) with the following relation:

$$J_s = -J \cdot \cos\beta + I \cdot \sin\beta \tag{1.2}$$

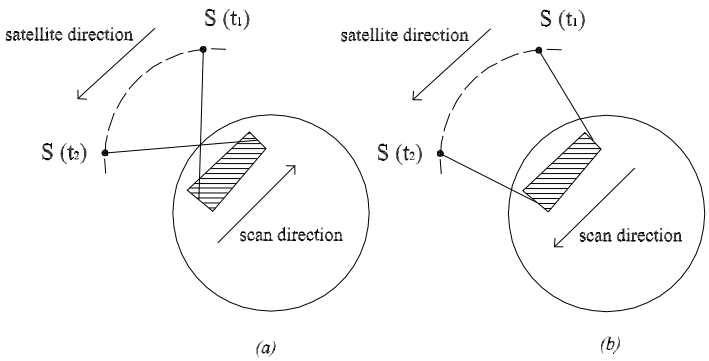


Figure 1.9: Forward scan mode (a), Reverse scan mode (b)

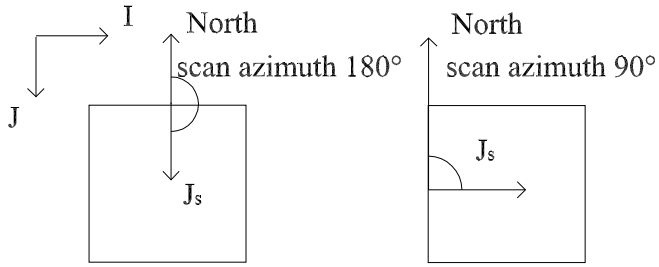


Figure 1.10: Example of J_s-axis direction

where β is the scan azimuth and I, J are the image coordinates.

1.4 Coordinate System Transformations

The global rotation matrix from the inertial geocentric system (ECI) to the Sensor system S can be expressed through three rotations [4]:

$$\mathbf{R}_{SI} = \mathbf{R}_{SB} \cdot \mathbf{R}_{BF} \cdot \mathbf{R}_{FI} \tag{1.3}$$

which can be detailed recalling the standard from the matrices representing

the rotations around the axes of the right-handed Cartesian coordinate system:

$$\mathbf{R}_X(\alpha) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & sen\alpha \\ 0 & -sen\alpha & \cos\alpha \end{vmatrix} \tag{1.4}$$

$$\mathbf{R}_Y(\beta) = \begin{vmatrix} \cos\beta & 0 & -sen\beta \\ 0 & 1 & 0 \\ sen\beta & 0 & \cos\beta \end{vmatrix} \tag{1.5}$$

$$\mathbf{R}_Z(\gamma) = \begin{vmatrix} \cos\gamma & sen\gamma & 0 \\ -sen\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{vmatrix} \tag{1.6}$$

(rotations α, β, γ are positive if they are counter clock wise when seen from the positive semi axes).

Inertial-Orbital matrix ($\mathbf{R}_{OI}$): this allows the passage from inertial geocentric system (ECI) to the orbital one. It is a function of right ascension of ascending node (Ω) and of orbit inclination (i):

$$\mathbf{R}_{OI} = \mathbf{R}_X(\mathrm{i}) \cdot \mathbf{R}_Z(\Omega) \tag{1.7}$$

Inertial-Flight matrix ($\mathbf{R}_{FI}$): this allows the passage from the inertial geocentric system (ECI) to the Flight one F. It is a function of Keplerian orbital parameters and varies with the time inside each scene (for each image row J):

$$\mathbf{R}_{FI} = \mathbf{R}_X(-\frac{\pi}{2}) \cdot \mathbf{R}_Z(\frac{\pi}{2}) \cdot \mathbf{R}_Z(U) \cdot \mathbf{R}_X(\mathrm{i}) \cdot \mathbf{R}_Z(\Omega) \tag{1.8}$$

where i is the inclination, Ω the right ascension of the ascending node, $U = \omega + \nu$ with ω argument of the perigee and ν true anomaly

Flight-Body matrix ($\mathbf{R}_{BF}$): it allows the passage from the Flight system F to the Body one B through the attitude angles (φ, ϑ, ψ) which depend on time (for each pixel row):

$$\mathbf{R}_{BF} = \mathbf{R}_Z(\psi) \cdot \mathbf{R}_Y(\vartheta) \cdot \mathbf{R}_X(\varphi) \tag{1.9}$$

Body-Sensor matrix ($\mathbf{R}_{SB}$): t allows the passage from the Body to the Sensor system. This matrix considers the deviation of the parallelism between axes $(X, Y, Z)_S$ and $(X, Y, Z)_B$ and it is considered constant during a scene for each sensor; the elements of the matrix may be provided in the metadata files. It happens for QuickBird and WorldView-1, for which the elements are supplied in form of quaternions $[q_1, q_2, q_3, q_4]$ describing the attitude of the camera coordinate system in the spacecraft body system; $\mathbf{R}_{SB}$ an identity matrix for EROS A, while

it is described by one rotation of yaw axis in yaw-roll plane with a tilt of - 5 degrees for Aft camera and +26 degrees for Fore camera for Cartosat-1 platform.

The rotation matrix for the transformation from ECI system to ECEF system ($\mathbf{R}_{EI}$) can be subdivided into four sequential steps, considering the motions of the Earth in space: precession, the secular change in the orientation of the Earth's rotation axis and the vernal equinox (described by the matrix $\mathbf{R}_P$); nutation, the periodic and short-term variation of the equator and the vernal equinox (described by the matrix $\mathbf{R}_N$); polar motion, the coordinates of the rotation axis relative to the IERS Reference Pole (described by the matrix $\mathbf{R}_M$); and Earth's rotation about its axis (described by the Sideral Time through the matrix $\mathbf{R}_S$) [7].

$$\mathbf{R}_{EI} = \mathbf{R}_M \cdot \mathbf{R}_S \cdot \mathbf{R}_N \cdot \mathbf{R}_P \tag{1.10}$$

The product of $\mathbf{R}_{EI}$ and $\mathbf{R}_{SI}^T$ matrices allows the passage from sensor S to ECEF system, with the final rotation matrix being:

$$\mathbf{R}_{ES} = \mathbf{R}_{EI} \cdot \mathbf{R}_{SI}^T = \mathbf{R}_Z(K) \cdot \mathbf{R}_Y(P) \cdot \mathbf{R}_X(W) \tag{1.11}$$

where the angles (K, P, W) define the satellite attitude at the epoch of the acquisition of image row J with respect to the ECEF system.

1.5 Interior Orientation and Self-Calibration Parameters

The interior orientation parameters describe the intrinsic geometric features of the sensor. Moreover, self-calibration parameters are used to correct the geometric errors in the CCD linear array and the optical system. Note that, referring to the linear array design, one CCD line can be made of ns segments being N_{pi} is the number of pixels contained in i-th segment. The modeling of the geometric errors is easy if carried out in the sensor system (S). For the sake of simplicity, here we consider models under the assumption of ns=1. More details may be found in [1].

1.5.1 CCD Linear Array Geometric Errors

The geometric errors that may occur in CCD linear array sensors are briefly described and modeled hereafter:

1. The change of the pixel size. It has the effect to change the image scale (Fig.1.11).

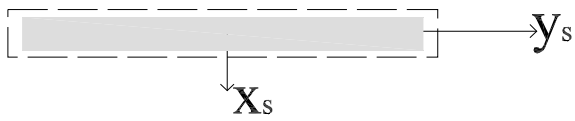


Figure 1.11: Effect of pixel size change in y_S direction "from [1]"

If (p_x, p_y) are the pixel dimensions and (dp_x, dp_y) a change of the pixel size in x direction and in y direction respectively, the errors (dx_p, dy_p) result:

$$dx_p = dp_x dy_p = y \cdot \frac{dp_y}{p_y} \tag{1.12}$$

The error dy_p may be also due to the focal length variation and the radial distortion. Note that the first order terms in (1.12), (1.15) are highly correlated. Therefore, it is not possible to estimate both the pixel dimension variation together the focal length variation.

2. The shifts and rotations of the CCD segments in the focal plane. These errors are described and modelled as follows:

   - shifts in x- and y-directions (Figs.1.12a, 1.12b): constant quantity dx_c and dy_c, respectively

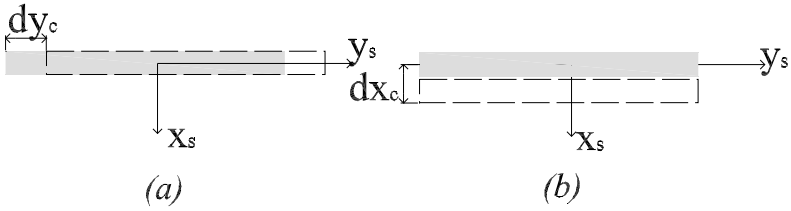


Figure 1.12: Shift of CCD segment in y_S direction (a) and shift of CCD segment in x_S direction (b) "from [1]"

   - effect of horizontal rotation θ in the CCD plane. The rotation produces the error dy_θ in y_S direction and dx_θ in x_S direction (Fig.1.13), but only the latter has to be considered since θ is small:

$$dx_\theta = y \cdot \sin\theta \tag{1.13}$$

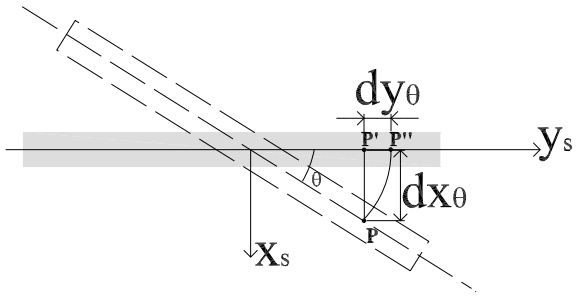


Figure 1.13: Effects of rotation of CCD segment in the focal plane "from [1]"

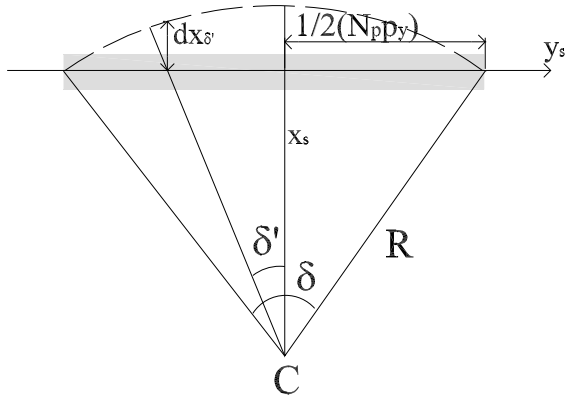


Figure 1.14: Line bending in the focal plane "from [1]"

3. The line bending in the focal plane. The straight CCD line is supposed to be deformed into an arc if the size of the bending is described by the central angle δ that subtends the arc described by the deformed line and the central angle δ' is related to the generic pixel position, the error results (Fig.1.14),

$$dx_{\delta'} = R\left(\cos\delta' - \cos\frac{\delta}{2}\right) \tag{1.14}$$

where $R = \frac{N_p \cdot p_y}{2sen\frac{\delta}{2}}$ and the bending is supposed in the plane defined by $x_S > 0$

If the CCD line is bending in the plane where $x_S < 0$, equation (1.14) is valid with the opposite sign.

1.5.2 Optical System Errors

The possible errors that may occur in an optical systems have been deeply investigated in close range, airborne and satellite photogrammetry [8], [9], [10].

1. The displacement of the lens principal point. This error is modelled with constant shifts Δx_p, Δy_p applied to the principal point coordinates (x_p, y_p) in x and y directions and is totally correlated with the shift of the CCD linear array in the focal plane.

2. The change of the focal length f. The effect of this error Δf in x and y directions is modelled as

$$\begin{aligned} dx_f &= -\frac{\Delta f}{f} \cdot \bar{x}_p \\ dy_f &= -\frac{\Delta f}{f} \cdot \bar{y}_p \end{aligned} \tag{1.15}$$

   where $\bar{x}_p = x - x_p$ and $\bar{y}_p = y - y_p$

3. The symmetric lens distortion is described by the coefficients k_1 and k_2 and modelled as

$$\begin{aligned} dx_r &= \left(k_1 \cdot r^2 + k_2 \cdot r^4\right) \cdot \bar{x}_p \\ dy_r &= \left(k_1 \cdot r^2 + k_2 \cdot r^4\right) \cdot \bar{y}_p \end{aligned} \tag{1.16}$$

   where $r^2 = \bar{x}_p^2 + \bar{y}_p^2$. For pushbroom sensors with CCD linear array only dy_r may be significant and $r^2 \cong \bar{y}_p^2$

4. The decentering lens distortion is modeled as

$$\begin{aligned} dx_d &= p_1 \cdot (r^2 + 2\bar{x}_p^2) + 2p_2 \cdot \bar{x}_p \bar{y}_p \\ dy_d &= 2p_1 \cdot \bar{x}_p \bar{y}_p + p_2 \cdot r^2 + 2\bar{y}_p^2 \end{aligned} \tag{1.17}$$

   Again, only dy_d may be significant.

1.5.3 Final Consideration about the Self-Calibration Parameters

Accounting for the described general models for the CCD linear array geometric errors and for the optical system errors and considering their significance and correlations, only a few self-calibration parameters need to be introduced.

Moreover, these parameters can be conveniently chosen to directly model self-calibration with respect to the image coordinates (I, J). In detail, the following self-calibration parameters are worth to be set up:

1. The position of principal point (I_0, J_0), accounting also for the shift of the CCD linear array in the focal plane (as described in Fig.1.12).
2. The rotation of the CCD linear array in the focal plane (k) (as described in Fig.1.13).
3. The change of the focal length (δf), accounting also for the scale variation and the isotropic change of pixel dimension.
4. The symmetric lens distortion up to the third order in y direction only (d_L).

Another possible parameters is δ representing the line bending, however, it will not be considered in the following rigorous model implementation.

1.6 Rigorous Model for Original Images (level 1A)

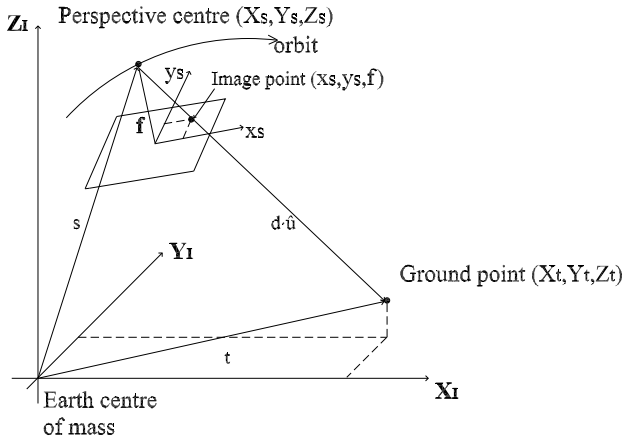


Figure 1.15: Central projection model with Sensor system and ECI system

As mentioned before, a rigorous model is based on the collinearity equations and describes the imagery acquisition both from the geometrical and physical (and stochastic) points of view. It is now possible to write the collinearity equations relating to the position of a point in the image space (Image point) to the corresponding point in the object space (Ground point), according to a central projection. In our case, the collinearity equations may be conveniently expressed in the Earth Centered Inertial (ECI) system starting from the relationship (Fig.1.15)

$$\begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix}_I = \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix}_I + d \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}_I \quad \Rightarrow \quad \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}_I = \frac{1}{d} \begin{bmatrix} X_t - X_s \\ Y_t - Y_s \\ Z_t - Z_s \end{bmatrix}_I \tag{1.18}$$

where $(X_t, Y_t, Z_t)_I$ are the ECI coordinates of the ground point (GP), $(X_s, Y_s, Z_s)_I$ are the ECI coordinates of the satellite (perspective center); $(u_x, u_y, u_z)_I$ are the components in the ECI system of the unit vector $\hat{u}$ directed from satellite to GP, and d is the distance from the perspective center to GP.

$$d = \sqrt{(X_t - X_s)^2 + (Y_t - Y_s)^2 + (Z_t - Z_s)^2} \tag{1.19}$$

Therefore, introducing the sensor system the collinearity equations read:

$$\frac{1}{d_s}\begin{bmatrix} x_S \\ y_S \\ f \end{bmatrix} = \mathbf{R}_{SI}\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}_I \quad \Rightarrow \quad \begin{bmatrix} x_S \\ y_S \\ f \end{bmatrix} = \frac{d_s}{d}\mathbf{R}_{SI}\begin{bmatrix} X_t - X_s \\ Y_t - Y_s \\ Z_t - Z_s \end{bmatrix} \tag{1.20}$$

where f is the focal distance, $\mathbf{R}_{SI}$ is the rotation matrix from ECI to sensor system and d_s is the perspective center to image point distance

$$d_s = \sqrt{x_S^2 + y_S^2 + f^2} \tag{1.21}$$

Finally, the standard form of two collinearity equations for each ground point is obtained by dividing the first two equations of (1.20) by the third one:

$$\begin{cases} \frac{x_S}{f} = \frac{\mathbf{R}_{SI,11}|X_t - X_s| + \mathbf{R}_{SI,12}|Y_t - Y_s| + \mathbf{R}_{SI,13}|Z_t - Z_s|}{\mathbf{R}_{SI,31}|X_t - X_s| + \mathbf{R}_{SI,32}|Y_t - Y_s| + \mathbf{R}_{SI,33}|Z_t - Z_s|} \\ \frac{y_S}{f} = \frac{\mathbf{R}_{SI,21}|X_t - X_s| + \mathbf{R}_{SI,22}|Y_t - Y_s| + \mathbf{R}_{SI,23}|Z_t - Z_s|}{\mathbf{R}_{SI,31}|X_t - X_s| + \mathbf{R}_{SI,32}|Y_t - Y_s| + \mathbf{R}_{SI,33}|Z_t - Z_s|} \end{cases} \tag{1.22}$$

With simple geometric consideration (Fig.1.1*a*) the collinearity equations can be written as a function of image coordinates, interior orientation and self-calibration parameters previously set up, Keplerian orbital and attitude parameters

$$\begin{cases} \frac{x_S}{f} = \tan\beta = \frac{d_{pix}}{f}\left[J - \mathrm{int}(J) - 0.5 - J_0 - k(I - I_0)\right] \\ \frac{y_S}{f} = \tan\alpha = -\frac{d_{pix}}{f}\left\{(I - I_0) + d_L(I - I_0)^3 + k\left[J - \mathrm{int}(J) - 0.5 - J_0\right]\right\} \end{cases} \tag{1.23}$$

Therefore, equating (1.22) and (1.23) the relationship between the image and the ground coordinates is found:

$$\begin{aligned} &F_1 = \frac{d_{pix}}{f}\left[J - \mathrm{int}(J) - 0.5 - J_0 - k(I - I_0)\right] + \\ &- \frac{\mathbf{R}_{SI,11}|X_t - X_s| + \mathbf{R}_{SI,12}|Y_t - Y_s| + \mathbf{R}_{SI,13}|Z_t - Z_s|}{\mathbf{R}_{SI,31}|X_t - X_s| + \mathbf{R}_{SI,32}|Y_t - Y_s| + \mathbf{R}_{SI,33}|Z_t - Z_s|} = 0 \\ &F_2 = -\frac{d_{pix}}{f}\left\{(I - I_0) + d_L(I - I_0)^3 + k\left[J - \mathrm{int}(J) - 0.5 - J_0\right]\right\} + \\ &- \frac{\mathbf{R}_{SI,21}|X_t - X_s| + \mathbf{R}_{SI,22}|Y_t - Y_s| + \mathbf{R}_{SI,23}|Z_t - Z_s|}{\mathbf{R}_{SI,31}|X_t - X_s| + \mathbf{R}_{SI,32}|Y_t - Y_s| + \mathbf{R}_{SI,33}|Z_t - Z_s|} = 0 \end{aligned} \tag{1.24}$$

The collinearity equations depend on the parameters described in the Sections 1.2, 1.3 and 1.5. In some cases the image metadata file is not supplied with the satellite position and velocity at every interval time; thereby the orbit is reconstructed using the procedure described in the Section 1.7.1 [11].

As mentioned, the approximate values for all parameters may be derived from the information contained in the metadata files, released together with the image ($a, i, \Omega, e, \nu, \omega, \varphi, \vartheta, \psi, f$ and I_0) or they are simply fixed to zero (a_i, b_i, c_i, J_0, k

and d_L). In theory, these approximate values must be corrected by an estimation process based on a suitable number of GCPs, for which the collinearity equations are written. Nevertheless, since the orbital arc related to each image acquisition is extremely short (a few hundreds of kilometers) compared to the whole orbit length (tens of thousands). Some Keplerian parameters are not estimable at all (a, e, ω) and others (i, Ω, T_P) are extremely correlated both among themselves and with respect to the sensor attitude, interior orientation and self-calibration parameters (f, I_0, J_0, k, d_L) [12]. The parameters estimable are $(a_i, b_i, c_i, T_P, f, I_0, J_0, k, d_L)$. Regarding the stochastic model, the standard deviations of the image observations are set equal, since manual measurement tests carried out independently by different operators range from $1/3$ to $1/2$ pixel in accuracy. For the GCP coordinates standard deviations are usually set equal to the mean values obtained during their direct surveying or cartographic selection [13].

In order to avoid instability due to high correlations among some parameters leading to design matrix pseudo-singularity, Singular Value Decomposition (SVD) and QR decomposition are employed to evaluate the actual rank of the design matrix, to select the actually estimable parameters and finally to solve the linearized collinearity equations system in the LS sense (see Chapter 3).

Moreover, the statistical significance of each estimable parameter is checked by a Student T-test so to avoid over-parameterization. In case of a statistically non-significant parameter, it is removed and the estimation process is repeated until all parameters are significant. Finally, only the really estimable and significant parameters are selected, acknowledging the well known parsimony principle for the functional models.

In Table 1.1 an example of selected parameters are presented; the tested image is acquired by EROS A, it was oriented varying the number of GCPs and for each orientation test the selected estimable and significant parameters are listed.

Pseudo-observation on the GCPs are also allowed, in order to account for their accuracy, which may vary depending on their source (cartography or direct surveying).

1.6.1 Atmospheric refraction effect

In this rigorous model also the refraction effect has been taken in account. This effect causes a well know bending of the optical paths due to the variation of atmospheric density (Fig.1.16*a*).

The purpose of atmospheric refraction modeling is the correction of the image coordinates in order to remove the refraction effect and to estimate the orientation parameters under the hypothesis of straight optical paths, which are actually modeled by collinearity equations (1.24).

Table 1.1: Parameters automatically selected by SISAR for ITA1-e1090724 - EROS A

GCP	SISAR parameters
9	a_0, a_1, b_0, b_1, c_0, T_P
13	a_0, a_1, b_0, b_1, c_0, T_P
17	a_0, a_1, b_0, b_1, c_0, f, a_2, b_2, c_1, c_2
21	a_0, a_1, b_0, b_1, c_0, f, b_2, c_2
25	a_0, a_1, b_0, b_1, c_0, f, T_P
29	a_0, a_1, b_0, b_1, c_0, f, b_2, c_1, c_2
33	a_0, a_1, b_0, b_1, c_0, f, b_2, c_1, c_2
37	a_0, a_1, b_0, b_1, c_0, f, b_2, c_1, c_2
41	a_0, a_1, b_0, b_1, c_0, f, b_2, c_2
45	a_0, a_1, b_0, b_1, c_0, f, b_2, c_2
49	a_0, a_1, b_0, b_1, c_0, f, b_2, c_2

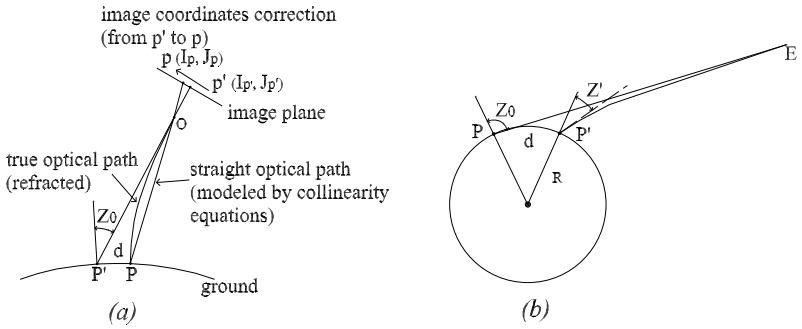
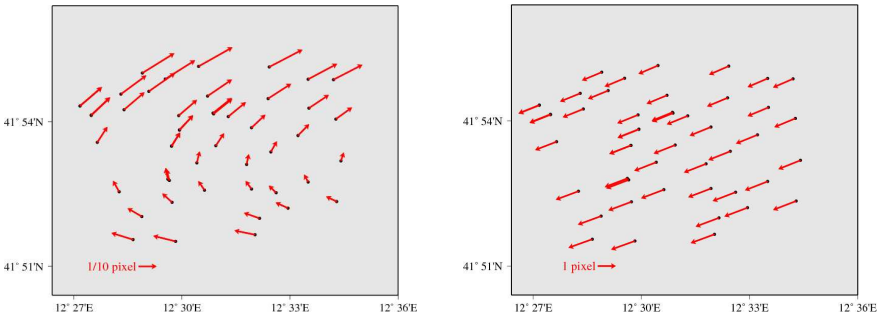


Figure 1.16: Atmospheric refraction model

In fact, the GP P is projected onto the image along a refracted path in p', but equations (1.24) model the collinearity condition along the straight path POp.

Therefore, we need to compute the correction from p' to p in order to properly adopt the model (1.24). The deviation from the collinearity assumption due to the atmospheric refraction is computed by a model duly described in ([2]) (Fig.1.16*b*, Tab.1.2) which basically allows to calculate the displacement d due to the refraction effect and the position of P' on the ground, starting from the



(a) EROS A Rome ITA1-e1038452 (γ=9.1°-9.4°)

(b) EROS A Rome ITA1-e1090724 (γ=31.0°- 40.1°)

Figure 1.17: Distribution of atmospheric refraction effect, represented by vector $\vec{a}$ computed for each GP

off-nadir angle.

Therefore, a first rough orientation is performed, neglecting the refraction effect, in order to estimate the off-nadir angle under which each GP is imaged, then the corresponding P'.

Moreover, through the collinearity equations, starting from P and P', the corresponding image position p and p' are computed, so that the components of the vector $\vec{a}$ suitable to remove the atmospheric refraction effect from the image coordinates can be computed by:

$$\begin{bmatrix} I_a \\ J_a \end{bmatrix} = \begin{bmatrix} I_p - I_{p'} \\ J_p - J_{p'} \end{bmatrix} \tag{1.25}$$

The vector $\vec{a}$ is applied to the coordinates collected on the image, from which a new estimation process is performed in order to refine the orientation parameters. Note that one iteration is usually enough, since the refraction is well estimated on the basis of the first rough orientation.

It is important to consider the refraction effect especially when the satellite attitude variation during the image acquisition causes a refraction effect not uniformly distributed (Fig.1.17*a*). On the contrary, its impact is lower when the satellite attitude variation causes a rather uniform refraction effect (Fig.1.17*b*).

Table 1.2: Displacement d over the local sphere approximating the ellipsoid due to the refraction effect function of z_0

z0 (°)	**z' (°)**	**z0 - z' (°)**	**d (m)**
10	9.9971	0.0029	0.47
20	19.9939	0.0061	1.06
30	29.9904	0.0096	1.97
40	39.986	0.014	3.62
45	44.9834	0.0166	5.03
50	49.9802	0.0198	7.21

1.6.2 Model computation

The functional model (for example, [14]) in short form reads: $f(y, x) = 0$ and its linearized form is $\mathbf{D}\boldsymbol{y}-\mathbf{A}\boldsymbol{x}-\boldsymbol{l}= 0$. In this model the two collinearity equations F_1 and F_2 (1.24) are written for each available GCP. As usual the solution is obtained iteratively due to non-linearity of the system; the iterative procedure stops when the estimated variance of unit weight reaches a minimum. At this step the variance of observations σ, generally set at $1/3$ to $1/2$ pixel level, is checked by the observation corrections (it could be overestimate or underestimate); if the relative difference between the variance applied for the solution (${\sigma^2}_{i-1}$) and the recomputed variance (${\sigma^2}_i$) is more than 5 percent, the average between two variances (σ^2) is used for a next cycle of iterations (1.26).

$$\sigma = \left| \frac{\sigma_{i-1} - \sigma_i}{\sigma_{i-1}} \right| \leq 5\% \tag{1.26}$$

If n is the number of GCPs, the relevant vectors and matrices can be sketched as follows:

vector of unknowns:

$$\boldsymbol{x}^T = |\boldsymbol{P}, \boldsymbol{PO}| \tag{1.27}$$

where $\boldsymbol{P}$ represent the correction of parameters

$$\boldsymbol{P} = |a_0, a_1, a_2, b_0, b_1, b_2, c_0, c_1, c_2, \delta T_P, \delta f, \delta I_0, J_0, k, d_L| \tag{1.28}$$

and $\boldsymbol{PO}$ represent the correction of ground coordinates

$$\boldsymbol{PO} = |\delta X_1, \delta Y_1, \delta Z_1, \cdots \delta X_n, \delta Y_n, \delta Z_n| \tag{1.29}$$

observation vector

$$\boldsymbol{y} = \left| \begin{array}{c} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_n \\ \boldsymbol{y}_{p1} \\ \vdots \\ \boldsymbol{y}_{pn} \end{array} \right| \text{ for i}^{\text{th}} \text{ GCP} \rightarrow \begin{cases} \boldsymbol{y}_i = \left| \begin{array}{c} I \\ J \end{array} \right| & \rightarrow \text{image coordinates} \\ \\ \boldsymbol{y}_{pi} = \left| \begin{array}{c} X \\ Y \\ Z \end{array} \right| & \rightarrow \text{ground coordinates} \end{cases} \tag{1.30}$$

design matrix

$$\mathbf{A} = \left| \begin{array}{ccccc} \mathbf{A}_1 & \mathbf{A}_{p1} & 0 & 0 & 0 \\ \mathbf{A}_2 & 0 & \mathbf{A}_{p2} & 0 & 0 \\ \vdots & 0 & 0 & \vdots & 0 \\ \mathbf{A}_n & 0 & 0 & 0 & \mathbf{A}_{pn} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{array} \right| \tag{1.31}$$

$$\text{for i}^{\text{th}} \text{ GCP} \rightarrow \quad \mathbf{A}_i^T = \left| \begin{array}{cc} \partial_{a_0} F_1 & \partial_{a_0} F_2 \\ \partial_{a_1} F_1 & \partial_{a_1} F_2 \\ \vdots & \vdots \\ \partial_{c_2} F_1 & \partial_{c_2} F_2 \\ \partial_{T_p} F_1 & \partial_{T_p} F_2 \\ \partial_f F_1 & \partial_f F_2 \\ \partial_{I_0} F_1 & \partial_{I_0} F_2 \\ \partial_{J_0} F_1 & \partial_{J_0} F_2 \\ \partial_k F_1 & \partial_k F_2 \\ \partial_{d_L} F_1 & \partial_{d_L} F_2 \end{array} \right| \quad \mathbf{A}_{pi} = \left| \begin{array}{ccc} \partial_{X_i} F_1 & \partial_{Y_i} F_1 & \partial_{Z_i} F_1 \\ \partial_{X_i} F_2 & \partial_{Y_i} F_2 & \partial_{Z_i} F_2 \end{array} \right| \tag{1.32}$$

matrix of observation and **pseudo-observation coefficients**

$$\mathbf{D} = \left| \begin{array}{cccc} \mathbf{D}_1 & & & 0 \\ & \mathbf{D}_2 & & \\ & & \ddots & \\ & & & \mathbf{D}_n \\ 0 & \cdots & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & \cdots & \cdots & 0 \end{array} \right| \begin{array}{l} \\ \\ \\ \\ \\ \\ \\ \end{array} \rightarrow \text{pseudo} - \text{observations}$$

$$\text{for i}^{\text{th}} \text{ GCP} \rightarrow \mathbf{D}_i = \left| \begin{array}{cc} \partial_I F_1 & \partial_J F_1 \\ \partial_I F_2 & \partial_J F_2 \end{array} \right| \tag{1.33}$$

known vector

$$\boldsymbol{l} = \left| \begin{array}{c} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \\ \vdots \\ \boldsymbol{l}_n \\ \boldsymbol{l}_{p1} \\ \boldsymbol{l}_{p2} \\ \vdots \\ \boldsymbol{l}_{pn} \end{array} \right| \text{for i}^{\text{th}} \text{ GCP} \rightarrow \left\{ \begin{array}{l} \boldsymbol{l}_i = \left| \begin{array}{c} \tilde{F}_1 \\ \tilde{F}_2 \end{array} \right| \\ \\ \\ \boldsymbol{l}_{pi} = \left| \begin{array}{c} X \\ Y \\ Z \end{array} \right| \end{array} \right. \tag{1.34}$$

1.6.3 Stochastic model

The stochastic model is $\mathbf{C}_{YY} = \sigma_0^2 \mathbf{Q}$, where $\sigma_0^2 = 1$ (a priori choice) and the cofactor matrix is chosen diagonal both for the observations and pseudo-observation $(I, J; E, N, h)$ for the sake of simplicity, being usually hard to evaluate

the correlations:

$$\mathbf{C}_{YY} = \mathbf{Q} = \begin{vmatrix} \mathbf{C}_{IJ}^{1} & & 0 & & & \\ & \ddots & & & & \\ 0 & & \mathbf{C}_{IJ}^{n} & & & \\ & & & \mathbf{C}_{XYZ}^{1} & & 0 \\ & & & & \ddots & \\ & & & 0 & & \mathbf{C}_{XYZ}^{n} \end{vmatrix} \tag{1.35}$$

$$\text{for i}^{\text{th}}\text{ GCP} \rightarrow \begin{cases} \mathbf{C}_{IJ}^{i} = \begin{vmatrix} \sigma_I^2 & 0 \\ 0 & \sigma_J^2 \end{vmatrix} \\ \mathbf{C}_{XYZ}^{i} = \begin{vmatrix} \sigma_{X_T}^2 & \sigma_{X_T}\sigma_{Y_T} & \sigma_{X_T}\sigma_{Z_T} \\ \sigma_{Y_T}\sigma_{X_T} & \sigma_{Y_T}^2 & \sigma_{Y_T}\sigma_{Z_T} \\ \sigma_{Z_T}\sigma_{X_T} & \sigma_{Z_T}\sigma_{Y_T} & \sigma_{Z_T}^2 \end{vmatrix} \end{cases} \tag{1.36}$$

The standard deviations of the image observations are set equal, considering that manual measurement tests carried out independently by different operators showed that an accuracy ranging from $1/3$ to $1/2$ pixel in image coordinates may be routinely achieved. For the GCP coordinates standard deviations are usually set equal to mean values obtained during their direct surveys or cartographic selection.

1.7 Rigorous Model for Pre-Processed Images (level 1B)

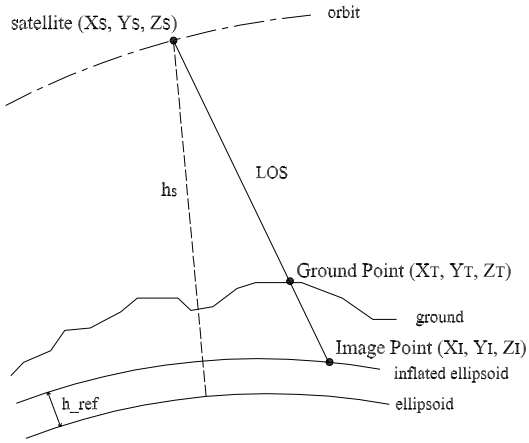


Figure 1.18: Model geometry

In this case it has to be noted that the images are projected onto a specific object (usually an "inflated" ellipsoid, derived from the WGS84 choosing a certain ellipsoidal height) (level 1B). The collinearity equations link points on the ground and points projected on the mentioned "inflated" ellipsoid [15] (Fig.1.18).

Each point on the ground surface corresponds to a point on "inflated" ellipsoid, identified from line of sight (LOS), i.e. the line directed from the perspective centre to the point on the ground. The collinearity condition is satisfied when $\hat{u}_{SI}$ (the unit vector directed from perspective centre to image point) coincides with $\hat{u}_{ST}$ (the unit vector directed from perspective centre to ground point), i.e., ground point and image point are lined up on LOS. The collinearity equations may be conveniently expressed in the ECEF system in vector form:

$$\hat{u}_{SI} = \mathbf{R} \cdot \hat{u}_{ST} \tag{1.37}$$

where $\mathbf{R}$ is a rotation matrix. In fact, relative "small" translation of ground with respect to ellipsoid can be expressed with an infinitesimal rotation around the perspective center, because the height of satellite platform (h_S) is much more than the difference of elevation between ground surface and the "inflated" ellipsoid (Δh). Under this infinitesimal rotation hypothesis ($\cos\varphi, \vartheta, \psi \cong 1$,

$\sin\varphi \cong \varphi, \sin\vartheta \cong \vartheta, \sin\psi \cong \psi$) the rotation matrix $\mathbf{R}$ is reduced to the sum of the unit matrix and an anti-symmetric matrix.

$$\mathbf{R} = \mathbf{I} + \delta\mathbf{R} = \begin{bmatrix} 0 & \varphi & \theta \\ -\varphi & 0 & \psi \\ -\theta & -\psi & 0 \end{bmatrix} \Rightarrow \mathbf{R} = \begin{bmatrix} 1 & \varphi & \theta \\ -\varphi & 1 & \psi \\ -\theta & -\psi & 1 \end{bmatrix} \tag{1.38}$$

where the attitude angles are supposed to be modelled by a time-dependent function up to the second order, similar to (1.1). The (1.37) can also be expressed in the following way:

$$\begin{bmatrix} X_I - X_s \\ Y_I - Y_s \\ Z_I - Z_s \end{bmatrix} = \rho\mathbf{R} \begin{bmatrix} X_T - X_s \\ Y_T - Y_s \\ Z_T - Z_s \end{bmatrix} \tag{1.39}$$

where

- ϱ is the scale factor, (ratio of perspective centre-image point distance d_{SI} and perspective centre-ground point distance d_{ST}: $\rho = d_{SI}/d_{ST}$)
- X_T, Y_T, Z_T are the ground coordinates in the ECEF system
- X_I, Y_I, Z_I are the image coordinates in the ECEF system
- X_s, Y_s, Z_s are the perspective centre coordinates in the ECEF system

Finally, note that in this case the collinearity equations on the basis of previous consideration, now reads:

$$\begin{cases} F_1 = \frac{|X_I - X_s|}{|Z_I - Z_s|} - \frac{\mathbf{R}_{SE,11}|X_T - X_s| + \mathbf{R}_{SE,12}|Y_T - Y_s| + \mathbf{R}_{SE,13}|Z_T - Z_s|}{\mathbf{R}_{SE,31}|X_T - X_s| + \mathbf{R}_{SE,32}|Y_T - Y_s| + \mathbf{R}_{SE,33}|Z_T - Z_s|} = 0 \\ F_2 = \frac{|Y_I - Y_s|}{|Z_I - Z_s|} - \frac{\mathbf{R}_{SE,21}|X_T - X_s| + \mathbf{R}_{SE,22}|Y_T - Y_s| + \mathbf{R}_{SE,23}|Z_T - Z_s|}{\mathbf{R}_{SE,31}|X_T - X_s| + \mathbf{R}_{SE,32}|Y_T - Y_s| + \mathbf{R}_{SE,33}|Z_T - Z_s|} = 0 \end{cases} \tag{1.40}$$

The collinearity equations depend on the parameters described in Sections 1.2 and 1.3. The approximate values for all parameters may be derived from the information contained in the metadata files, released together with the image ($\alpha, \beta, e, i, \Omega$) or they are simply fixed to zero (a_i, b_i, c_i). Again, these approximate values must be corrected by an estimation process based on a suitable number of GCPs, for which the collinearity equations are written. Finally, the parameters estimable are (a_i, b_i, c_i). It has to be noted that the image coordinates in the collinearity equations (1.40) must be expressed in the ECEF system, while on the other hand the image coordinates are obtained by point measurement on the image so that only I and J are known. First of all, it is required to change image

coordinates into cartographic coordinate through the simple equations:

$$\begin{cases} N_P = N_A - J \cdot p \\ E_P = E_A + I \cdot p \end{cases} \tag{1.41}$$

where

- N_P, E_P are the North and East UTM WGS84 coordinates of a generic point P
- N_A, E_A are the North and East UTM WGS84 coordinates of upper left corner A of the image
- I, J are the coordinates of a generic point P in the image system
- p is the pixel size (in meters)

Moreover, the cartographic coordinates are converted into geographic coordinates (latitude φ and longitude λ). Finally, since the ellipsoidic height (h) of points on the image is the elevation of "inflated" ellipsoid, the geodetic coordinate (φ, λ, h) are converted in Cartesian coordinates (ECEF system).

1.7.1 Computation of Satellite Positions

In general the detailed information about the satellite position are not supplied for the level 1B images, therefore the satellite coordinates can be computed only on the basis of the angles (azimuth α and elevation e) that define satellite position with respect to image center (Fig.1.7).

Referring to Fig.1.19 the off-nadir angle γ and B distance between the image center C and the satellite position S are calculated with the sine theorem.

$$\sin\gamma = \frac{(R + h\_ref) \cdot \sin(e + \pi/2)}{(R + H)} \tag{1.42}$$

$$B = \frac{(R + h\_ref) \cdot \sin\omega}{\sin\gamma} \tag{1.43}$$

where R is the radius of local sphere, H is the height of satellite, h_{ref} is the height of "inflated" ellipsoid in reference to the WGS84 ellipsoid, e is the elevation angle and $\omega = \pi - (\gamma + e + \pi/2)$.

The satellite coordinates in a geodetic local system, whose origin is the center of image (C) are:

$$\begin{cases} X_{S(C)} = B \cdot \cos e \cdot \sin\alpha \\ Y_{S(C)} = B \cdot \cos e \cdot \cos\alpha \\ Z_{S(C)} = B \cdot \sin e \end{cases} \tag{1.44}$$

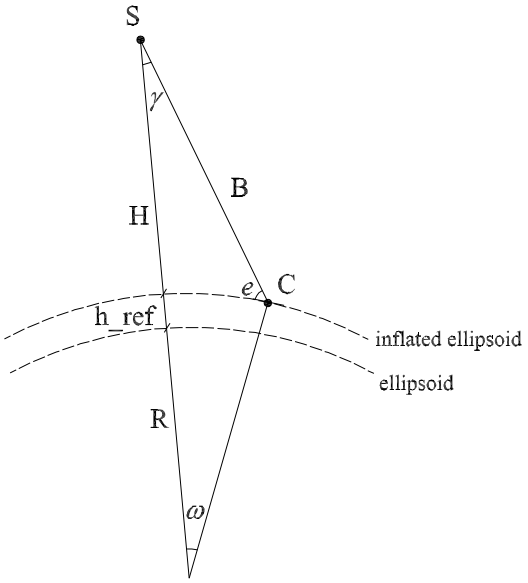


Figure 1.19: Satellite position with respect to image center

Local coordinates are transformed into ECEF coordinates, and from the unique satellite position it is possible to reconstruct the orbit segment. Satellite coordinates are converted from ECEF to ECI system (1.10), then from ECI to the orbital system with $\mathbf{R}_{OI}$ matrix (1.7). Generally the inclination i is known, on the contrary the right ascension of ascending node Ω can be calculated under the hypothesis of Keplerian orbit ($Z_{SOrbit} = 0$). This equation has two solutions (Ω_1, Ω_2) that correspond to ascending and descending orbit respectively. If the satellite moves on descending orbit while acquires images, the right value of Ω is obtained with $X_{S(C)} < 0$ (abscissa in the orbital system) (Fig.1.20*a*), otherwise the solution is with $X_{S(C)} > 0$ if the satellite is ascending (Fig.1.20*b*). Since the orbital arc reletad to the one image acquisition is extremaly short, a circular approximation of the orbit is allowed. On the orbital plane the satellite position relative to image's center ($S_{(C)}$) identifies the U_C angle, where $U_C = \arctan(Y_{S(C)}/X_{S(C)})$; consequently the satellite position relative to each image row is obtained, moving the satellite forward and backward on the orbit with respect to central position:

$$U_i = U_C - (Js_C - Js_i) \cdot \Delta U \tag{1.45}$$

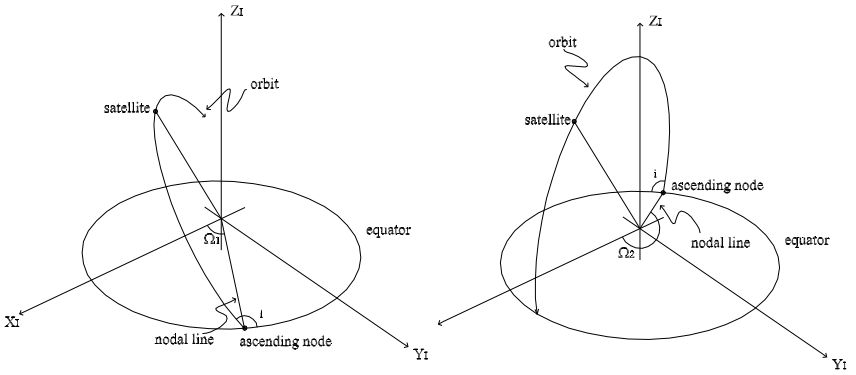


Figure 1.20: Representation of orbits relative to the two solution 1 (a) and 2 (b)

where Js_C is the scanning row of image's center, and Js_i is a generic scanning row of the image and ΔU is the angular displacement relative to one scanning row.

$$\Delta U = \frac{ATR}{R + h\_ref} \tag{1.46}$$

The satellite position (S_i) in the orbital system is:

$$\begin{cases} X_{S(i)} = (R+H) \cdot \cos U_i \\ Y_{S(i)} = (R+H) \cdot \sin U_i \\ Z_{S(i)} = 0 \end{cases} \tag{1.47}$$

where R is the radius of local sphere and H is height of satellite platform (Fig.1.21). Then the satellite coordinates must be converted into the ECI system and into the ECEF system. The ECEF coordinates enter in collinearity equations.

Sometimes the information about the elevation angle e and the azimuth angle α, that described the satellite position in the Local system with the origin in the centre of the image, are very rough so that the satellite is positioned on the orbit at about some hundred meters respect to the right position during the image acquisition; consequently the satellite position is estimated in a more accurate waythrough a Direct Linear Transformation (DLT).

The DLT establishes a rough relation between image coordinates and ground

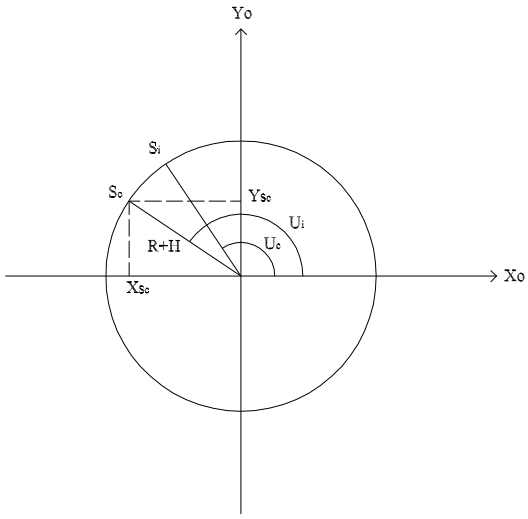


Figure 1.21: Satellite coordinates in Orbital system

coordinates in a Cartesian Local system. It is based on the following equations:

$$\begin{aligned} I &= \frac{L_1 \cdot E + L_2 \cdot N + L_3 \cdot U + L_4}{L_9 \cdot E + L_{10} \cdot N + L_{11} \cdot U + 1} \\ J &= \frac{L_5 \cdot E + L_6 \cdot N + L_7 \cdot U + L_8}{L_9 \cdot E + L_{10} \cdot N + L_{11} \cdot U + 1} \end{aligned} \tag{1.48}$$

where (I, J) are the image coordinates, (E, N, U) are the ground coordinates respect to the Cartesian Local system centered in the center of the image and the L_i are the DLT parameters.

Starting from some GCPs coordinates (minimum 7 GCPs), the DLT parameters are estimated (see equation 1.48); satellite position in the Cartesian Local system, related to the image center, is computed using the DLT parameters for a fixed height of the satellite.

Local coordinates are transformed into ECEF coordinates, and from the unique satellite position it is possible to reconstruct the orbit segment. Due to the short lenght of the orbital arc related to the image acquisition, it is possible to approximate it with an arc of circumference.

This procedure for satellite position computation is used for Cartosat-1 platform. The image acquired is not pre-processed but few information regarding satellite features are supplied in metadata file that do not permit to recostruct the orbital segment.

1.7.2 Model computation

The functional model (for example, [14]) in short form reads: $f(y,x)=0$ and its linearized form is $\mathbf{D}\boldsymbol{y}-\mathbf{A}\boldsymbol{x}-\boldsymbol{l}=0$. In this model the two collinearity equations F_1 and F_2 (1.40) are written for each available GCP:

$$\begin{array}{l} F_1 = \frac{X_I-X_s}{Z_I-Z_s} - \frac{(X_I-X_s)+\varphi\cdot(Y_I-Y_s)+\theta\cdot(Z_I-Z_s)}{-\theta\cdot(X_I-X_s)-\psi\cdot(Y_I-Y_s)+(Z_I-Z_s)} = 0 \\ F_2 = \frac{Y_I-Y_s}{Z_I-Z_s} - \frac{-\varphi\cdot(X_I-X_s)+(Y_I-Y_s)+\psi\cdot(Z_I-Z_s)}{-\theta\cdot(X_I-X_s)-\psi\cdot(Y_I-Y_s)+(Z_I-Z_s)} = 0 \end{array} \tag{1.49}$$

If n is the number of GCPs, the relevant vectors and matrices can be sketched as follows:

vector of unknowns:

$$\boldsymbol{x}^T = |\boldsymbol{P}, \boldsymbol{PO}| \tag{1.50}$$

$$\boldsymbol{P} = |a_0, b_0, c_0, a_1, b_1, c_1, a_2, b_2, c_2| \tag{1.51}$$

and $\boldsymbol{PO}$ represent the correction of ground coordinates (see 1.29).

Regards **observation vector**, **matrix of observation** and **known vector** they have the same form of level 1A model (see equations 1.30, 1.33 and 1.34).

design matrix

$$\mathbf{A} = \begin{vmatrix} \mathbf{A}_1 & \mathbf{A}_{p1} & 0 & 0 & 0 \\ \mathbf{A}_2 & 0 & \mathbf{A}_{p2} & 0 & 0 \\ \vdots & 0 & 0 & \vdots & 0 \\ \mathbf{A}_n & 0 & 0 & 0 & \mathbf{A}_{pn} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} \tag{1.52}$$

$$\text{for i}^{\text{th}}\text{ GCP} \rightarrow \mathbf{A}_i^T = \begin{vmatrix} \partial_{a_0}F_1 & \partial_{a_0}F_2 \\ \partial_{b0}F_1 & \partial_{b0}F_2 \\ \partial_{c0}F_1 & \partial_{c0}F_2 \\ \partial_{a_1}F_1 & \partial_{a_1}F_2 \\ \partial_{b1}F_1 & \partial_{b1}F_2 \\ \partial_{c1}F_1 & \partial_{c1}F_2 \\ \partial_{a2}F_1 & \partial_{a2}F_2 \\ \partial_{b2}F_1 & \partial_{b2}F_2 \\ \partial_{c2}F_1 & \partial_{c2}F_2 \end{vmatrix} \quad \mathbf{A}_{pi} = \begin{vmatrix} \partial_{X_i}F_1 & \partial_{Y_i}F_1 & \partial_{Z_i}F_1 \\ \partial_{X_i}F_2 & \partial_{Y_i}F_2 & \partial_{Z_i}F_2 \end{vmatrix} \tag{1.53}$$

1.7.3 Stochastic model

The stochastic model is $\mathbf{C}_{YY} = \sigma_0^2 \mathbf{Q}$, where $\sigma_0^2 = 1$ (a priori choice) and the cofactor matrix is chosen diagonal both for the observations and pseudo-observation $(I, J; E, N, h)$ for the sake of simplicity, being usually hard to evaluate the correlations

$$\mathbf{C}_{YY} = \begin{vmatrix} \mathbf{C}^1_{XYZ_I} & & 0 & & & \\ & \ddots & & & & \\ 0 & & \mathbf{C}^n_{XYZ_I} & & & \\ & & & \mathbf{C}^1_{XYZ_T} & & 0 \\ & & & & \ddots & \\ & & & 0 & & \mathbf{C}^n_{XYZ_T} \end{vmatrix} \tag{1.54}$$

where

$$\text{for i}^{\text{th}}\text{ GCP} \rightarrow \begin{cases} \mathbf{C}_{XYZ_I} = \begin{vmatrix} \sigma^2_{X_I} & \sigma_{X_I}\sigma_{Y_I} & \sigma_{X_I}\sigma_{Z_I} \\ \sigma_{Y_I}\sigma_{X_I} & \sigma^2_{Y_I} & \sigma_{Y_I}\sigma_{Z_I} \\ \sigma_{Z_I}\sigma_{X_I} & \sigma_{Z_I}\sigma_{Y_I} & \sigma^2_{Z_I} \end{vmatrix} \\ \mathbf{C}_{XYZ_T} = \begin{vmatrix} \sigma^2_{X_T} & \sigma_{X_T}\sigma_{Y_T} & \sigma_{X_T}\sigma_{Z_T} \\ \sigma_{Y_T}\sigma_{X_T} & \sigma^2_{Y_T} & \sigma_{Y_T}\sigma_{Z_T} \\ \sigma_{Z_T}\sigma_{X_T} & \sigma_{Z_T}\sigma_{Y_T} & \sigma^2_{Z_T} \end{vmatrix} \end{cases} \tag{1.55}$$

These matrices are obtained from covariance matrices of image coordinates and GPS coordinates expressed in Geodetic local system through a rotation in ECEF system.

$$\begin{cases} \mathbf{C}_{NEh_I} = \begin{vmatrix} \sigma^2_{NE_I} & 0 & 0 \\ 0 & \sigma^2_{NE_I} & 0 \\ 0 & 0 & \sigma^2_{h_I} \end{vmatrix} \\ \mathbf{C}_{NEh_T} = \begin{vmatrix} \sigma^2_{NE_T} & 0 & 0 \\ 0 & \sigma^2_{NE_T} & 0 \\ 0 & 0 & \sigma^2_{h_T} \end{vmatrix} \end{cases} \tag{1.56}$$

Because image cartographic coordinates are achieved with (1.41) equations, the value of σ_{NE_I} comes from the collimation accuracy (σ_{IJ}): through variance propagation based on (1.39)

$$\sigma_{NE_I} = p \cdot \sigma_{IJ} \tag{1.57}$$

On the contrary, since the height of the "inflated" ellipsoid is fixed, σ_{h_I} can be fixed to zero. The values of σ_{NE_T} and σ_{h_T} represent the horizontal and vertical accuracies of the GPS coordinates.

1.8 Stereo Rigorous Model

The rigorous model developed to orientate both level 1A and level 1B single scene can be extended to manage both along-track and across-track stereopairs [16]. In this case, it has to be noted that the orbital elements are the same for two images if they were acquired during the same orbital path (along-track stereopairs) or they are different if the images are acquired during two different orbital paths at different epochs (across-track stereopairs). Again, the approximate values of these parameters can be computed by using the information in the metadata file and have to be corrected by a LS estimation process based on a suitable number of GCPs. The estimable parameters can be selected using the procedure described in chapter 3. In this respect, since the tie points (TPs) may be conveniently considered, it is necessary to establish a procedure for the computation of their approximate ground coordinates, which have to be LS estimated together with all other parameters.

First of all, it is useful to orientate separately the single scenes adopting the already described rigorous model. These separated orientations have to be considered just as preliminary ones; they have to be refined in a block adjustment possibly including suitable TPs. In this case, the question arise to compute the approximate position of TPs. In theory, after the preliminary orientations, the homologous rays should intersect, identifying a unique ground point for each couple of homologous points chosen over the image. Nevertheless, errors remaining in the separate orientations cause the well known parallaxes, so that homologous rays do not intersect and no ground point can be found by intersection. Therefore, it is necessary to set up a rule to compute the approximate TPs ground positions.

To this aim, the choice was made to compute the positions of the two points on the homologous rays at minimal distance and then to average their coordinates (Fig.1.22).

The minimum distance between the two rays is computed. The equations of two rays supposed straight can be written in parametric form

$$r_1 : \begin{vmatrix} X_1 \\ Y_1 \\ Z_1 \end{vmatrix} = \begin{vmatrix} X_{01} \\ Y_{01} \\ Z_{01} \end{vmatrix} + s \cdot \begin{vmatrix} \mathrm{a}_1 \\ \mathrm{b}_1 \\ \mathrm{c}_1 \end{vmatrix} \qquad r_2 : \begin{vmatrix} X_2 \\ Y_2 \\ Z_2 \end{vmatrix} = \begin{vmatrix} X_{02} \\ Y_{02} \\ Z_{02} \end{vmatrix} + t \cdot \begin{vmatrix} \mathrm{a}_2 \\ \mathrm{b}_2 \\ \mathrm{c}_2 \end{vmatrix} \quad (1.58)$$

where $(X_0, Y_0, Z_0)_{1,2}$ are the coordinates of perspective centers in the ECEF system for the two images, $(a, b, c)_{1,2}$ are direction cosine known from the separate orientations. The condition to identify the two points on the rays at minimal

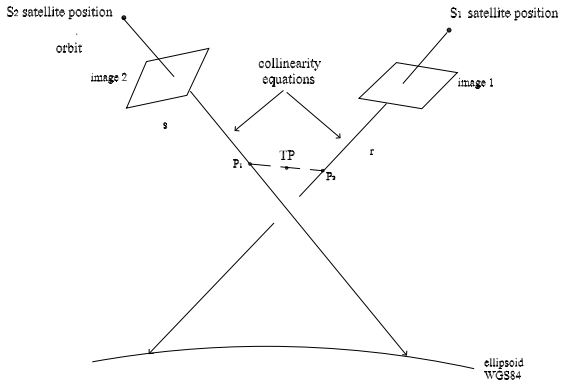


Figure 1.22: TP determination

distance reads:

$$\begin{cases} \frac{\partial d^2(s,t)}{\partial t} = 0 \\ \frac{\partial d^2(s,t)}{\partial s} = 0 \end{cases} \text{ with } \quad d^2(s,t) = (X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2 \quad (1.59)$$

what leads to:

$$\begin{vmatrix} s \\ t \end{vmatrix}_{MIN} = \begin{vmatrix} A & B \\ C & D \end{vmatrix}^{-1} \cdot \begin{vmatrix} E \\ F \end{vmatrix} \Rightarrow$$

$$\begin{cases} A = (a_1^2 + b_1^2 + c_1^2) \\ B = -(a_1 a_2 + b_1 b_2 + c_1 c_2) \\ C = -(a_1 a_2 + b_1 b_2 + c_1 c_2) \\ D = (a_2^2 + b_2^2 + c_2^2) \\ E = (X_{02} - X_{01}) \cdot a_1 + (Y_{02} - Y_{01}) \cdot b_1 + (Z_{02} - Z_{01}) \cdot c_1 \\ F = (X_{02} - X_{01}) \cdot a_2 + (Y_{02} - Y_{01}) \cdot b_2 + (Z_{02} - Z_{01}) \cdot c_2 \end{cases} \quad (1.60)$$

$$\begin{vmatrix} X_1 \\ Y_1 \\ Z_1 \end{vmatrix}_{MIN} = \begin{vmatrix} X_{01} \\ Y_{01} \\ Z_{01} \end{vmatrix} + s_{MIN} \cdot \begin{vmatrix} \mathrm{a}_1 \\ \mathrm{b}_1 \\ \mathrm{c}_1 \end{vmatrix} \quad \begin{vmatrix} X_2 \\ Y_2 \\ Z_2 \end{vmatrix}_{MIN} = \begin{vmatrix} X_{02} \\ Y_{02} \\ Z_{02} \end{vmatrix} + t_{MIN} \cdot \begin{vmatrix} \mathrm{a}_2 \\ \mathrm{b}_2 \\ \mathrm{c}_2 \end{vmatrix} \quad (1.61)$$

$$\begin{vmatrix} \tilde{X}_{TP} \\ \tilde{Y}_{TP} \\ \tilde{Z}_{TP} \end{vmatrix} = \frac{1}{2} \cdot \begin{vmatrix} X_1 + X_2 \\ Y_1 + Y_2 \\ Z_1 + Z_2 \end{vmatrix}_{\mathrm{MIN}} \quad (1.62)$$

1.8.1 Model computation

The functional model (for example, [14]) in short form reads: $f(y,x) = 0$ and its linearized form is $\mathbf{D}\boldsymbol{y} - \mathbf{A}\boldsymbol{x} - \boldsymbol{l} = 0$. In this model the four collinearity equations F_1 and F_2 for the first image and F_3 and F_4 for the second image (1.22) are written for each available GPs, both GCPs and TPs. In the following model the all unknowns related to both images have to be estimated; it is valid for across-track stereopairs while for along-track stereopairs a unique value for the internal orientation and self-calibration parameters is estimated, since the two images are acquired by the same sensor with a small time delay.

vector of unknowns:

$$\boldsymbol{x}^T = \left| \boldsymbol{P}^{(1)}, \boldsymbol{P}^{(2)}, \ \boldsymbol{PO} \ \right| \tag{1.63}$$

$$\boldsymbol{P}^{(i)} = \left| a_0^{(i)}, a_1^{(i)}, a_2^{(i)}, b_0^{(i)}, b_1^{(i)}, b_2^{(i)}, c_0^{(i)}, c_1^{(i)}, c_2^{(i)}, \delta T_P^{(i)}, \delta f^{(i)}, \delta I_0^{(i)}, J_0^{(i)}, k^{(i)}, d_L^{(i)} \right| \tag{1.64}$$

where i is referred to the first or the second image and PO represents the correction of ground coordinates, both GCPs and TPs (see 1.29). The approximate TPs coordinates are computed using (1.62).

observation vector

$$\boldsymbol{y} = \left| \begin{array}{c} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_n \\ \boldsymbol{y}_{p1} \\ \vdots \\ \boldsymbol{y}_{pn} \end{array} \right| \text{ for i}^{\text{th}} \text{ GCP} \rightarrow \begin{cases} \boldsymbol{y}_i = \left| \begin{array}{c} I^{(1)} \\ J^{(1)} \\ I^{(2)} \\ I^{(2)} \end{array} \right| & \rightarrow \text{image coordinates} \\ \\ \boldsymbol{y}_{pi} = \left| \begin{array}{c} X \\ Y \\ Z \end{array} \right| & \rightarrow \text{ground coordinates} \end{cases} \tag{1.65}$$

known vector

$$\boldsymbol{l}^T = \left| \begin{array}{cccccccccccc} \boldsymbol{l}_1^{(1)} & \boldsymbol{l}_2^{(1)} & \dots & \boldsymbol{l}_n^{(1)} & \boldsymbol{l}_1^{(2)} & \boldsymbol{l}_2^{(2)} & \dots & \boldsymbol{l}_n^{(2)} & \boldsymbol{l}_{p1} & \boldsymbol{l}_{p2} & \dots & \boldsymbol{l}_{pn} \end{array} \right|$$

$$\text{for i}^{\text{th}}\ \text{GP} \rightarrow \begin{cases} \boldsymbol{l}_i^T(1) = \left| \begin{array}{cc} \tilde{F}_1 & \tilde{F}_2 \end{array} \right| \\ \boldsymbol{l}_i^T(2) = \left| \begin{array}{cc} \tilde{F}_3 & \tilde{F}_4 \end{array} \right| \\ \boldsymbol{l}_{pi}^T = \left| \begin{array}{ccc} X & Y & Z \end{array} \right| \end{cases} \tag{1.66}$$

matrix of observation

$$\mathbf{D} = \left| \begin{array}{ccc} \mathbf{D}_1^{(1)} & & 0 \\ & \mathbf{D}_2^{(1)} & \\ & \ddots & \\ 0 & & \mathbf{D}_n^{(1)} \\ \mathbf{D}_1^{(2)} & & 0 \\ \vdots & \mathbf{D}_2^{(2)} & \vdots \\ 0 & \ddots & \\ & & \mathbf{D}_n^{(2)} \\ 0 & & 0 \\ \vdots & & \vdots \\ 0 & & 0 \end{array} \right| \begin{array}{l} \\ \\ \rightarrow \text{first image} \\ \\ \\ \rightarrow \text{second image} \\ \\ \\ \rightarrow \text{pseudo} - \text{observations} \\ \\ \\ \end{array} \tag{1.67}$$

$$\text{for i}^{\text{th}}\ \text{GP} \rightarrow \mathbf{D}_{\text{i}}^{(1)} = \left| \begin{array}{cccc} \partial_{I^{(1)}} F_1 & \partial_{J^{(1)}} F_1 & 0 & 0 \\ \partial_{I^{(1)}} F_2 & \partial_{I^{(1)}} F_2 & 0 & 0 \end{array} \right|$$

$$\text{for i}^{\text{th}}\ \text{GP} \rightarrow \mathbf{D}_{\text{i}}^{(2)} = \left| \begin{array}{cccc} 0 & 0 & \partial_{I^{(2)}} F_3 & \partial_{J^{(2)}} F_3 \\ 0 & 0 & \partial_{I^{(2)}} F_3 & \partial_{I^{(2)}} F_4 \end{array} \right|$$

design matrix

$$\mathbf{A} = \left| \begin{array}{ccccc} \mathbf{A}_1^{(1)} & \mathbf{A}_{p1}^{(1)} & 0 & 0 & 0 \\ \mathbf{A}_2^{(1)} & 0 & \mathbf{A}_{p2}^{(1)} & 0 & 0 \\ \vdots & 0 & 0 & \vdots & 0 \\ \mathbf{A}_n^{(1)} & 0 & 0 & 0 & \mathbf{A}_{pn}^{(1)} \\ \mathbf{A}_1^{(2)} & \mathbf{A}_{p1}^{(2)} & 0 & 0 & 0 \\ \mathbf{A}_2^{(2)} & 0 & \mathbf{A}_{p2}^{(2)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}_n^{(2)} & 0 & 0 & 0 & \mathbf{A}_{pn}^{(2)} \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & 0 \end{array} \right|$$

$$\text{for i}^{\text{th}} \quad \text{GP} \rightarrow \left\{ \begin{array}{l} \mathbf{A}_i^{(1)} = \left| \begin{array}{ccccccc} \partial_{a_0^{(1)}} F_1 & \partial_{a_1^{(1)}} F_1 & \cdots & \partial_{d_L^{(1)}} F_1 & 0 & \cdots & 0 \\ \partial_{a_0^{(1)}} F_2 & \partial_{a_1^{(1)}} F_2 & \cdots & \partial_{d_L^{(1)}} F_2 & 0 & \cdots & 0 \end{array} \right| \\ \mathbf{A}_i^{(2)} = \left| \begin{array}{ccccccc} 0 & \cdots & 0 & \partial_{a_0^{(2)}} F_3 & \partial_{a_1^{(2)}} F_3 & \cdots & \partial_{d_L^{(2)}} F_3 \\ 0 & \cdots & 0 & \partial_{a_0^{(2)}} F_4 & \partial_{a_1^{(2)}} F_4 & \cdots & \partial_{d_L^{(2)}} F_4 \end{array} \right| \end{array} \right.$$

$$\text{for i}^{\text{th}} \quad \text{GP} \rightarrow \left\{ \begin{array}{l} \mathbf{A}_{pi}^{(1)} = \left| \begin{array}{ccc} \partial_{X_i} F_1 & \partial_{Y_i} F_1 & \partial_{Z_i} F_1 \\ \partial_{X_i} F_2 & \partial_{Y_i} F_2 & \partial_{Z_i} F_2 \end{array} \right| \\ \mathbf{A}_{p_i}^{(2)} = \left| \begin{array}{ccc} \partial_{X_i} F_3 & \partial_{Y_i} F_3 & \partial_{Z_i} F_3 \\ \partial_{X_i} F_4 & \partial_{Y_i} F_4 & \partial_{Z_i} F_4 \end{array} \right| \end{array} \right. \tag{1.68}$$

This functional model is valid also for level 1B stereopairs, reminding that the unknowns are only attitude parameters and the image coordinates are expressed in ECEF system as described in Sect.1.7.2.

Chapter 2

Rational Polynomial Function with Rational Polynomial Coefficients

---

A few years ago high resolution satellite imagery were available to a limited number of government and defence agencies that managed such imagery with highly sophisticated software and hardware tools.

High resolution satellite imagery are now available in different formats and processing levels and at an affordable price. These types of sensors and their growing availability are revolutionizing the role of satellite imagery in a number of applications ranging from intelligency to insurance, media, marketing, agriculture, utilities, urban planning, forestry, environmental monitoring, transportation, real estate etc.

One of the primary barriers to a wider adaptation and utilization of satellite imagery is the sensor model being able to provide a high level geometric correction through the image orientation. Sensor models are a key component to represent the functional relationships between the image space and the object space, and are essential for single/multi imagery orientation.

Even if the rigorous models should theoretically provide the highest accuracy, they are only available for some satellites and can be managed by some commercial available software. Moreover, in order to estimate the unknown parameters of rigorous models, users are still faced with the challenging task of

recovering the exterior orientation of the sensor using a set of GCPs usually no small than 10. When no or few GCPs are available, users cannot recover the exterior orientation of the sensor and therefore are unable to perform various mapping and data collection operations. With the introduction of generalized sensor models, this situation has changed considerably. Generalized sensor models, such as the RPF [17], have smoothed the requirement to manage a physical sensor model. Furthermore, as the RPCs implicitly provides the interior and (approximate) exterior sensor orientation, the availability of several GCPs is no longer a mandatory requirement. Consequently, the use of the RPCs for photogrammetric mapping is becoming a new standard in high-resolution satellite imagery that has already been implemented in various high-resolution sensors, such as Ikonos, QuickBird, WorldView-1 and GeoEye-1.

2.1 RPCs Usage and Orientation Refinement

As mentioned before, some companies (for example DigitalGlobe for QuickBird and WorldView and Space Imaging for Ikonos and GeoEye-1, India Space Research Organization for Cartosat-1) usually supply the RPCs, as part of the image metadata to enable image orientation via RPF.

The RPF relate object point coordinates (latitude φ, longitude λ and height h) to pixel coordinates (I, J), as a physical sensor models, but in the form of ratios of polynomial expressions:

$$I = \frac{P_1(\varphi, \lambda, h)}{P_2(\varphi, \lambda, h)} \qquad J = \frac{P_3(\varphi, \lambda, h)}{P_4(\varphi, \lambda, h)} \tag{2.1}$$

where φ, λ are the geographic coordinates, h is the height above the WGS84 ellipsoid and (I, J) are the image coordinates. The order of these four polynomials is usually limited to 3 so that each polynomial takes the generic form:

$$P_n = \sum_{i=0}^{m_1} \sum_{j=0}^{m_2} \sum_{k=0}^{m_3} t_{ijk} \varphi^i \lambda^j h^k \tag{2.2}$$

with $0 \leq m_1 \leq 3$; $0 \leq m_2 \leq 3$; $0 \leq m_3 \leq 3$ and $m_1 + m_2 + m_3 \leq 3$, where t_{ijk} are the RPCs.

The number of RPCs depends obviously on the polynomial order: if the equations (2.1) are written with third order polynomials, the maximum number of coefficients is 80 (20 for each polynomial). Actually, the total number of RPCs is reduced to 78, because the two equations can be divided for the zero order

terms of the denominators; so that the final form of the equations (2.1) reads:

$$\begin{aligned} I &= \frac{P_1(\varphi,\lambda,h)}{P_2(\varphi,\lambda,h)} = \frac{a_0+a_1\lambda+a_2\varphi+a_3h+a_4\lambda\varphi+\ldots+a_{17}\lambda^3+a_{18}\varphi^3+a_{19}h^3}{1+b_1\lambda+b_2\varphi+b_3h+b_4\lambda\varphi+\ldots+b_{17}\lambda^3+b_{18}\varphi^3+b_{19}h^3} \\ J &= \frac{P_3(\varphi,\lambda,h)}{P_4(\varphi,\lambda,h)} = \frac{c_0+c_1\lambda+c_2\varphi+c_3h+c_4\lambda\varphi+\ldots+c_{17}\lambda^3+c_{18}\varphi^3+c_{19}h^3}{1+d_1\lambda+d_2\varphi+d_3h+d_4\lambda\varphi+\ldots+d_{17}\lambda^3+d_{18}\varphi^3+d_{19}h^3} \end{aligned} \tag{2.3}$$

where now a_j, b_j, c_j, d_j are the RPCs.

The great power of these equations is the independence from the physical characteristic of the image acquisition [18]. Although ground coordinates are not directly connected with the acquisition physics, it is possible taking into account the further approximated considerations [19]: ratios of the first order terms can represent distortions caused by the optical projection, while corrections such as Earth curvature, atmospheric refraction and lens distortion can be well modelled by the second order terms; other unknown and more complex distortions with high order components may be absorbed by the third order terms.

The ground coordinates (φ, λ, h) in the equation (2.1) are normalized to (-1, +1) range using normalization parameters supplied in the metadata file, in order to improve the numerical precision during the computation.

The generic simple formula utilized for the normalization, is:

$$T_n = \frac{T - T_{offset}}{T_{scale}} \tag{2.4}$$

where T_n are the normalized coordinates, T_{offset}, T_{scale} are the normalization parameters available in the metadata file and T is the original ground or image coordinate $(T = I, J; \varphi, \lambda, h)$.

Since the residual bias may be present into the RPCs, the orientation can be refined on the basis of the known GPs, acting as GCPs. A possible refinement of the model (2.1) (written in normalized coordinates), allowing for bias compensation, is accomplished in a quite common way with the introduction of a simple first order polynomial in the RPF (2.5) whose parameters are estimated, provided a suitable number of GCPs is known [20].

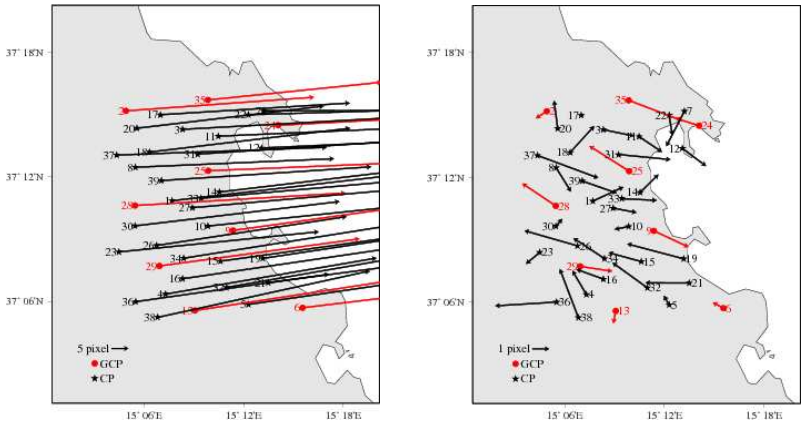
$$\begin{aligned} I_n &= A_0 + I_n \cdot A_1 + J_n \cdot A_2 + \frac{P_1(\varphi_n,\lambda_n,h_n)}{P_2(\varphi_n,\lambda_n,h_n)} = \\ &= A_0 + I_n \cdot A_1 + J_n \cdot A_2 + \frac{a_0+a_1\lambda_n+a_2\varphi_n+a_3h_n+a_4\lambda_n\varphi_n+\ldots+a_{17}\lambda_n^3+a_{18}\varphi_n^3+a_{19}h_n^3}{1+b_1\lambda_n+b_2\varphi_n+b_3h_n+b_4\lambda_n\varphi_n+\ldots+b_{17}\lambda_n^3+b_{18}\varphi_n^3+b_{19}h_n^3} \\ J_n &= B_0 + J_n \cdot B_1 + I_n \cdot B_2 + \frac{P_3(\varphi_n,\lambda_n,h_n)}{P_4(\varphi_n,\lambda_n,h_n)} = \\ &= B_0 + J_n \cdot B_1 + I_n \cdot B_2 + \frac{c_0+c_1\lambda_n+c_2\varphi_n+c_3h_n+c_4\lambda_n\varphi_n+\ldots+c_{17}\lambda_n^3+c_{18}\varphi_n^3+c_{19}h_n^3}{1+d_1\lambda_n+d_2\varphi_n+d_3h_n+d_4\lambda_n\varphi_n+\ldots+d_{17}\lambda_n^3+d_{18}\varphi_n^3+d_{19}h_n^3} \end{aligned} \tag{2.5}$$

where (I_n, J_n) are the normalized images coordinates, and P_i are third order polynomial functions of object space normalized coordinates $(\varphi_n, \lambda_n, h_n)$; A_i and B_i terms describe image shift and drift effects in particular:

- $A_0, A_1, A_2, B_0, B_1, B_2$ describe a complete affine transformation
- A_0, A_1, B_0, B_1 model the shift and drift
- A_0, B_0, describe a simple coordinate shift

The six new coefficients (A_i, B_i) are LS estimated based on GCPs. It is noted that in theory the model is not linear, since the 2^{nd} and 3^{rd} terms of the right side involve both observations (I_n, J_n) and parameters (A_i, B_i).

Nevertheless, usually in the right side observations I_n, J_n are considered as fixed coefficients, so that the model is treated as linear with respect to the six coefficients (A_i, B_i) (Fig.2.1a, 2.1b).



(a) QuickBird Augusta *P001 - uncorrected image

(b) QuickBird Augusta *P001 - corrected image

Figure 2.1: Example of residuals adjustment with an affine transformation on a QuickBird image

The formal structure of the functional model with full parametrization for the final estimation is the following:

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{l} \tag{2.6}$$

where $\mathbf{A}$ is the **design matrix**

$$\mathbf{A} = \left| \begin{array}{cc} \mathbf{A}_1 & \rightarrow \quad 1°\text{GCP} \\ \mathbf{A}_2 & \rightarrow \quad 2°\text{GCP} \\ . & \\ \mathbf{A}_k & \rightarrow \quad k°\text{GCP} \end{array} \right| \text{for i}^{\text{th}} \text{ GCP} \rightarrow \mathbf{A}_{\text{t}} = \left| \begin{array}{cccccc} 1 & I_n & J_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & J_n & I_n \end{array} \right| \tag{2.7}$$

$\boldsymbol{l}$ is the **known term**

$$\boldsymbol{l} = \left| \begin{array}{cc} \boldsymbol{l}_1 & \rightarrow \quad 1°\text{GCP} \\ \boldsymbol{l}_2 & \rightarrow \quad 2°\text{GCP} \\ . & \\ \boldsymbol{l}_k & \rightarrow \quad k°\text{GCP} \end{array} \right| \text{for i}^{\text{th}} \text{ GCP} \rightarrow \boldsymbol{l}_{\text{t}} = \left| \begin{array}{c} \frac{P_1(\varphi_n,\lambda_n,h_n)}{P_2(\varphi_n,\lambda_n,h_n)} \\ \frac{P_3(\varphi_n,\lambda_n,h_n)}{P_4(\varphi_n,\lambda_n,h_n)} \end{array} \right| \tag{2.8}$$

$\boldsymbol{y}$ is the **observation vector**

$$\boldsymbol{y} = \left| \begin{array}{c} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_i \\ \vdots \\ \boldsymbol{y}_n \end{array} \right| \quad \text{for i}^{\text{th}} \text{ GCP} \quad \rightarrow \quad \boldsymbol{y}_i = \left| \begin{array}{c} I \\ J \end{array} \right| \tag{2.9}$$

$\boldsymbol{x}$ is the **unknown vector**

$$\boldsymbol{x}^T = |A_0, A_1, A_2, B_0, B_1, B_2| \tag{2.10}$$

As regards the stochastic model, an unit identity diagonal cofactor matrix for observation (I, J) is assumed.

2.2 RPCs Generation

The RPCs can be generated by terrain-dependent scenario without using physical sensor model [21] or according to a terrain-independent scenario, using known physical sensor model.

For the terrain-dependent scenario, which is a kind of registration of the original image to 3D geometry represented by the GCPs, the RPF model tries to approximate the complicated imaging geometry across the image scene using polynomial terms. The solution is highly dependent on the actual terrain relief, the distribution and the number of GCPs. The RPCs have to be estimated in a LS adjustment so that the number of GCPs could be very high (at least 39 if RPCs up to the third order are looked for). This method is very weak and vulnerable in presence of outliers and it is likely to cause deformations far from the GCPs returning not good accuracies. Therefore, the RPFs solved by terrain-dependent approach must not be used as a replacement sensor model if high accuracy is required [21], [22], [23] and will not be considered anymore hereafter.

2.2.1 RPCs generation for terrain-independent scenario

For a terrain-independent scenario, a 2D image grid covering the full extent of the image is established and its corresponding 3D object grid with several layers (e.g., four or more layers for the third-order case) slicing the entire elevation range is generated. The horizontal coordinates (X, Y) of a point of the 3D object grid are calculated from a point (I, J) of the image grid using the physical sensor model with an a priori selected elevation Z. Then, the RPCs are LS estimated with the object grid points and the image grid points. This terrain-independent computational scenario can make the RPF model a good replacement to the physical sensor models, and has been widely used to determine the RPCs.

It has to be underlined that in the usually adopted terrain-independent approach, the least square solution is often carried out through a regularization, since unknown RPCs may be highly correlated so that the design matrix is almost rank deficient [24]. In order to overcome the regularization requirements, an innovative algorithm for the RPCs extraction, with a terrain independent approach, is analyzed. In details, at first an image discretization is made, dividing the full extent image space in a 2D grid. Then, the points of the 2D image grid are used to generate the 3D ground grid: the image was oriented and by the knowledge of the rigorous orientation sensor model, the collinearity equations were derived and used to create the 3D grid, starting from each point of the 2D grid image. In this respect, it has to be underlined that the 2D grid is actually a regular grid, whereas the 3D one is not strictly regular, due to the image attitude. Moreover, the 3D grid points were generated intersecting the straight lines modelled by the collinearity equations with surfaces (approximately ellipsoids)

concentric to the WGS84 ellipsoid, placed at regular elevation steps. So, the dimension of the 3D grid is both based on the full extent of the image and the elevation range of the terrain.

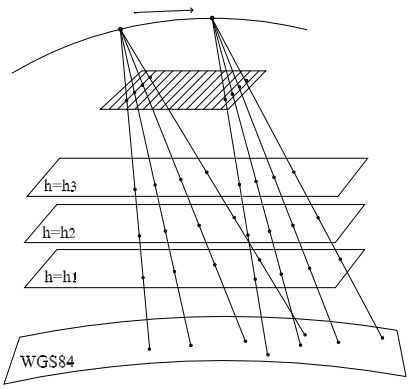


Figure 2.2: Grid for RPCs generation in the terrain-independent approach

The grid contains several elevation layers uniformly distributed, and the points on one layer have the same elevation value (Fig.2.2).

Note that the finest subdivision depends on the incompressible error of the rigorous model used to generate the RPCs, so that a very fine discretization is unuseful and an upper discretization limit also exists. The RPCs least squares estimation [22] is based on the linearization of the generic RPFs equations, which can be written as:

$$\begin{aligned} I_n + b_1\lambda_n I_n + b_2\varphi_n I_n + ... + b_{19}h_n^3 I_n - a_0 - a_1\lambda_n - a_2\varphi_n ... - a_{19}h_n^3 = 0 \\ J_n + d_1\lambda_n J_n + d_2\varphi_n J_n + ... + d_{19}h_n^3 J_n - c_0 - c_1\lambda_n - c_2\varphi_n ... - c_{19}h_n^3 = 0 \end{aligned} \tag{2.11}$$

where a_i, b_i, c_i, d_i are the RPCs (78 coefficients for third order polynomials), (I_n, J_n) and $(\varphi_n, \lambda_n, h_n)$ are the normalized coordinates obtained throughout the equation (2.4), with scale and offset factors computed according to:

$$\begin{cases} w_{offset} = \min(w_k) \\ w_{scale} = \max(w_k) - \min(w_k) \end{cases} where \quad w = \varphi, \lambda, h$$
$$\begin{cases} I_{offset} = J_{offset} = 1 \\ I_{scale} = n^{\circ}column - 1 \\ J_{scale} = n^{\circ}row - 1 \end{cases} \tag{2.12}$$

where k is the number of available GCPs and n° column/row are the overall columns/rows of the image; the normalization range is (0, 1).

Since the equations (2.11) are completely independent, the least squares estimation may be performed separately for the two image coordinates (I_n, J_n):

$$\begin{aligned} \mathbf{A}_I \boldsymbol{x}_I + \boldsymbol{y}_I = 0 \\ \mathbf{A}_J \boldsymbol{x}_J + \boldsymbol{y}_J = 0 \end{aligned} \tag{2.13}$$

where $\mathbf{A}_I$, $\mathbf{A}_J$ are the design matrices, $\boldsymbol{x_I}$,$\boldsymbol{x_J}$ are the unknown parameters (RPCs) and $\boldsymbol{y_I}$,$\boldsymbol{y_J}$ are the known terms (image coordinates):

$$\mathbf{A}_I = \left| \begin{array}{c|cl} \mathbf{A}_{I,1} & \rightarrow & 1^{\circ}\text{GCP} \\ \mathbf{A}_{I,2} & \rightarrow & 2^{\circ}\text{GCP} \\ \vdots & & \\ \mathbf{A}_{I,k} & \rightarrow & k^{\circ}\text{GCP} \end{array} \right| \quad \boldsymbol{x}_I = \left| \begin{array}{c} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{19} \\ b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{19} \end{array} \right| \quad \boldsymbol{y}_I = \left| \begin{array}{c|cl} I_{n,1} & \rightarrow & 1^{\circ}\text{GCP} \\ I_{n,2} & \rightarrow & 2^{\circ}\text{GCP} \\ \vdots & & \\ I_{n,k} & \rightarrow & k^{\circ}\text{GCP} \end{array} \right| \tag{2.14}$$

for a generic GCP l the design matrix row $\mathbf{A}_{I,l}$ has form that is:

$$\begin{array}{rl} \mathbf{A}_{I,l} = | & I_n\varphi_n \quad I_n\lambda_n \quad I_nh_n \quad \cdots \quad I_n\varphi_n^3 \quad I_n\lambda_n^3 \quad I_nh_n^3 \quad -1 \\ & -\varphi_n \quad -\lambda_n \quad -h_n \quad \cdots \quad -\varphi_n^3 \quad -\lambda_n^3 \quad -h_n^3 \; | \end{array} \tag{2.15}$$

$$\mathbf{A}_J = \left| \begin{array}{l} \mathbf{A}_{J,1} \\ \mathbf{A}_{J,2} \\ \vdots \\ \mathbf{A}_{J,k} \end{array} \right| \begin{array}{l} \rightarrow \quad 1°\text{GCP} \\ \rightarrow \quad 2°\text{GCP} \\ \\ \rightarrow \quad k°\text{GCP} \end{array} \qquad \boldsymbol{x}_J = \left| \begin{array}{c} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{19} \\ d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{19} \end{array} \right| \qquad \boldsymbol{y}_J = \left| \begin{array}{l} J_{n,1} \\ J_{n,2} \\ \vdots \\ J_{n,k} \end{array} \right| \begin{array}{l} \rightarrow \quad 1°\text{GCP} \\ \rightarrow \quad 2°\text{GCP} \\ \\ \rightarrow \quad k°\text{GCP} \end{array} \tag{2.16}$$

for a generic GCP l the design matrices row $\mathbf{A}_{J,l}$ has the form that is:

$$\begin{array}{l} \mathbf{A}_{J,l} = |J_n\varphi_n \quad J_n\lambda_n \quad J_nh_n \quad \cdots \quad J_n\varphi_n^3 \quad J_n\lambda_n^3 \quad J_nh_n^3 \quad -1 \\ \qquad\quad -\varphi_n \quad -\lambda_n \quad -h_n \quad \cdots \quad -\varphi_n^3 \quad -\lambda_n^3 \quad -h_n^3 \; | \end{array} \tag{2.17}$$

Deeper investigations underlined that many RPCs are highly correlated. In order to avoid instability due to high correlations, leading to a pseudo-singular design matrix, Tickhonov regularization is usually used. Generally, the regularization is exploited in a Tickhonov fashion, adopting a damping factor to the diagonal of the normal matrix, in order to guarantee its non singularity. A new alternative approach is based on the Singular Value Decomposition (SVD) and QR decomposition which are employed to evaluate the actual rank of the design matrix and to select the actual estimable coefficients [25], [13]; again, the SVD based subset selection procedure is due to Golub, Klema and Stewart [26], [27].

In details, at first it is necessary to select the RPCs estimable from the observations for the system of linear equations ($\mathbf{A}\boldsymbol{x}=\boldsymbol{l}$), with $\mathbf{A} \in \Re^{m \cdot n}$ (with $m \geq n$); the remaining coefficients need to be constrained to their initial (approximate) values, which are fixed to zero for all the coefficients.

The SVD is computed and used both to calculate the approximate values of RPCs to normalize the design matrix $\mathbf{A}$ and to determine its actual rank r; the threshold used to evaluate r is based on the ratio between the maximum and the allowed minimum singular values; reference values are $10^{-4} \div 10^{-5}$ [28].

Moreover, the statistical significance of each estimable coefficient is checked by a Student T-test, so to avoid overparametrization; in case of not statistically significant coefficient, it is removed and the estimation process is repeated until all coefficients are significant. In most of the cases the "degrees of freedom" are high (more than 100), thus there could be considered infinite, converting the

Student-T distribution in a normal standard distribution.

Atmosferic refraction effect is taken into account in RPCs generation; in details a correction vectors $\vec{a}_j$, relating to j^{th} point of 2D grid, is added to the coordinates of j^{th} 2D image grid point; this vector $\vec{a}_j$ is computed as weighted average of the correction vectors attaining to the GCPs ($\vec{b}_0$, $\vec{b}_i$, $\vec{b}_n$) (n is the GCPs number), being the weights the 2D distances of the 2D image grid point from all the GCP images ($d_{j_{2D}}$-i_{GCP}: distance from the j^{th} 2D grid point (j_{2D}) to i^{th} GCP (i_{GCP})) (2.18)

$$\vec{a}_j = \frac{\vec{b}_1 \cdot \frac{1}{d_{(j_{2D}-1_{GCP})}} + \vec{b}_i \cdot \frac{1}{d_{(j_{2D}-i_{GCP})}} + \cdots + \vec{b}_n \cdot \frac{1}{d_{(j_{2D}-n_{GCP})}}}{\frac{1}{d_{(j_{2D}-1_{GCP})}} + \frac{1}{d_{(j_{2D}-i_{GCP})}} + \cdots + \frac{1}{d_{(j_{2D}-n_{GCP})}}} \tag{2.18}$$

2.2.2 RPCs generation starting from vendors RPCs

When the physical sensor model is not available the RPCs can be generated starting from the RPCs supplied together the metadata file, by Companies that manage the different sensors. A general advantage of generating RPCs through the estabilished procedure, is to work with a remarkably reduced number of RPCs, only those are significant. An additional advantage in generating user own RPCs from vendor RPCs raises when one image is supplied in separate tiles, each with its RPCs. In this case, RPCs for whole image can be produced. The usage of vendors RPCs to generate new RPCs is based on the costruction of terrein grid on different layers covering the interest entire elevation range. The relations which link the ground and image coordinates, espressed in (2.3), are used to compute the image coordinates using the 3D object grid points and vendors RPCs. The least squares estimation, combined with SVD and QR decompositions, are applied to estimate the new RPCs; they are also refined with a minimum number of GCP necessary to estimate a shift or an affine transformation.

2.3 Stereo Model via RPCs

RPF model represents an attractive tool also for managing stereopairs orientation and possible subsequent DSM generation. Also in this case, the first problem to face with it is the computation of the approximate coordinates on the ground (φ, λ, h), to inizialize the linearization of the observation equations used for their least squares estimation. A forward procedure is based on the Direct Linear Transformation (DLT). The near linear projection of the high resolution satellite image ensures rapid convergence of the spatial intersection from even very coarse initial values for the object point coordinates.
The DLT is not using any preinformation about image orientation. The 22 unknowns (11 for each images) for the transformation of the object coordinates to the image coordinates have to be determined with at least 6 GCPs. The DLT equations are:

$$\begin{array}{l} I = \frac{L_1 \cdot E + L_2 \cdot N + L_3 \cdot U + L_4}{L_9 \cdot E + L_{10} \cdot N + L_{11} \cdot U + 1} \\ J = \frac{L_5 \cdot E + L_6 \cdot N + L_7 \cdot U + L_8}{L_9 \cdot E + L_{10} \cdot N + L_{11} \cdot U + 1} \end{array} \tag{2.19}$$

where (I, J) are the image coordinates, (E, N, U) are the ground coordinates respect to the Cartesian Local system centered in the center of the image and the L_i are the DLT parameters. In case of stereopairs the equations (2.19) are doubled and expressed in the following forms for every GCP:

$$\begin{cases} \left(L_9^1 E + L_{10}^1 N + L_{11}^1 U + 1\right) \cdot I^{(1)} - \left(L_1^1 E + L_2^1 N + L_3^1 U + L_4^1\right) = 0 \\ \left(L_9^1 E + L_{10}^1 N + L_{11}^1 U + 1\right) \cdot J^{(1)} - \left(L_5^1 E + L_6^1 N + L_7^1 U + L_8^1\right) = 0 \\ \left(L_9^2 E + L_{10}^2 N + L_{11}^2 U + 1\right) \cdot I^{(2)} - \left(L_1^2 E + L_2^2 N + L_3^2 U + L_4^2\right) = 0 \\ \left(L_9^2 E + L_{10}^2 N + L_{11}^2 U + 1\right) \cdot J^{(2)} - \left(L_5^2 E + L_6^2 N + L_7^2 U + L_8^2\right) = 0 \end{cases} \tag{2.20}$$

where superscripts 1 or 2 are related to the first and the second image respectively.

The functional model used is $\mathbf{A}\boldsymbol{x}+\boldsymbol{l}=0$:

where $\mathbf{A}$ is the **design matrix**

$$\mathbf{A} = \left| \begin{array}{c} \mathbf{A}^1 \\ \mathbf{A}^2 \end{array} \right| \begin{array}{l} \rightarrow \\ \rightarrow \end{array} \begin{array}{l} \text{first image} \\ \text{second image} \end{array} \qquad \mathbf{A}^i = \left| \begin{array}{c} \mathbf{A}_1^i \\ \mathbf{A}_2^i \\ \vdots \\ \mathbf{A}_k^i \end{array} \right| \begin{array}{l} \rightarrow \\ \rightarrow \\ \\ \rightarrow \end{array} \begin{array}{l} 1°\text{GCP} \\ 2°\text{GCP} \\ \\ k°\text{GCP} \end{array}$$

$$\mathbf{A}_j^1 = \left| \begin{array}{cc} -E & 0 \\ -N & 0 \\ -U & 0 \\ -1 & 0 \\ E \cdot I^{(1)} & E \cdot J^{(1)} \\ N \cdot I^{(1)} & N \cdot J^{(1)} \\ U \cdot I^{(1)} & U \cdot J^{(1)} \\ 0 & -E \\ 0 & -N \\ 0 & -U \\ 0 & -1 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{array} \right| \qquad \mathbf{A}_j^2 = \left| \begin{array}{cc} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ -E & 0 \\ -N & 0 \\ -U & 0 \\ -1 & 0 \\ E \cdot I^{(2)} & E \cdot J^{(2)} \\ N \cdot I^{(2)} & N \cdot J^{(2)} \\ U \cdot I^{(2)} & U \cdot I^{(2)} \\ 0 & -E \\ 0 & -N \\ 0 & -U \\ 0 & -1 \end{array} \right| \tag{2.21}$$

$\boldsymbol{x}$ and $\boldsymbol{l}$ are the **unknown** and **known terms** respectively

$$\boldsymbol{x} = \left| \begin{array}{c} L_j^1 \\ L_j^2 \end{array} \right| \quad \boldsymbol{l} = \left| \begin{array}{c} \boldsymbol{l}^1 \\ \boldsymbol{l}^2 \end{array} \right| \begin{array}{l} \rightarrow \\ \rightarrow \end{array} \begin{array}{l} \text{first image} \\ \text{second image} \end{array} \qquad \boldsymbol{l}^i = \left| \begin{array}{c} I^i \\ J^i \\ I^i \\ J^i \\ \vdots \\ I^i \\ J^i \end{array} \right| \begin{array}{l} \\ \rightarrow 1°\text{GCP} \\ \\ \rightarrow 2°\text{GCP} \\ \\ \\ \rightarrow k°\text{GCP} \end{array} \tag{2.22}$$

L_j^i are the DLT parameters relative to i image.

The stochastic model is $\mathbf{C}_{YY} = \sigma_0^2 \mathbf{Q}$, where $\sigma_0^2 = 1$ (a priori choice) and the cofactor matrix is chosen as a identity diagonal matrix.

Once the DLT parameters are estimated, they can be applied to all the TPs identified in both images to compute their approximate Cartesian Local coordinates, which are then transformed into ellipsoidal ones. The functional model

($\mathbf{A}\boldsymbol{x}+\boldsymbol{l}=0$) for the estimation TPs Cartesian Local coordinates is the following, starting from (2.20):

$\mathbf{A}$ is the design matrix

$$\mathbf{A} = \left| \begin{array}{c|ll} \mathbf{A}^1 & \rightarrow & \text{first image} \\ \mathbf{A}^2 & \rightarrow & \text{second image} \end{array} \right| \qquad \mathbf{A}^i = \left| \begin{array}{c|ll} \mathbf{A}_1^i & \rightarrow & 1^\circ\text{GCP} \\ \mathbf{A}_2^i & \rightarrow & 2^\circ\text{GCP} \\ \vdots & & \\ \mathbf{A}_k^i & \rightarrow & k^\circ\text{GCP} \end{array} \right| \tag{2.23}$$

$$\text{for i}^{\text{th}}\text{ TP} \rightarrow \left\{ \begin{array}{l} \mathbf{A}_j^1 = \left| \begin{array}{ccc} (L_9^1 I - L_1^1) & (L_{10}^1 I - L_2^1) & (L_{11}^1 I - L_3^1) \\ (L_9^1 J - L_5^1) & (L_{10}^1 J - L_6^1) & (L_{11}^1 J - L_7^1) \end{array} \right| \\ \mathbf{A}_j^2 = \left| \begin{array}{ccc} (L_9^2 I - L_1^2) & (L_{10}^2 I - L_2^2) & (L_{11}^2 I - L_3^2) \\ (L_9^2 J - L_5^2) & (L_{10}^2 J - L_6^2) & (L_{11}^2 J - L_7^2) \end{array} \right| \end{array} \right.$$

$\boldsymbol{x}$ is the **unknown vector**

$$\boldsymbol{x}^T = \left| \begin{array}{ccc} E & N & U \end{array} \right| \tag{2.24}$$

$\boldsymbol{l}$ is the **known vector**

$$\boldsymbol{l} = \left| \begin{array}{c|l} \boldsymbol{l}^1 & \rightarrow \text{first image} \\ \boldsymbol{l}^2 & \rightarrow \text{second image} \end{array} \right| \quad \boldsymbol{l}^i = \left| \begin{array}{c|ll} \boldsymbol{l}_1^i & \rightarrow & 1^\circ\text{GCP} \\ \boldsymbol{l}_2^i & \rightarrow & 2^\circ\text{GCP} \\ \vdots & & \\ \boldsymbol{l}_k^i & \rightarrow & k^\circ\text{GCP} \end{array} \right| \quad \boldsymbol{l}_j^i = \left| \begin{array}{c} \left(L_4^i - I^i\right) \\ \left(L_8^i - J^i\right) \end{array} \right| \tag{2.25}$$

The approximate TPs are refined using the RPCs corrected with affine transformation. At first the affine parameters (6 for each images) are estimated through the equations (2.5) (two for each image), established for each available GCP.

$$\begin{array}{l} I_n = A_0 + I_n \cdot A_1 + J_n \cdot A_2 + \frac{P_1(\varphi_n,\lambda_n,h_n)}{P_2(\varphi_n,\lambda_n,h_n)} = \\ = A_0 + I_n \cdot A_1 + J_n \cdot A_2 + \frac{a_0+a_1\lambda_n+a_2\varphi_n+a_3h_n+a_4\lambda_n\varphi_n+\ldots+a_{17}\lambda_n^3+a_{18}\varphi_n^3+a_{19}h_n^3}{1+b_1\lambda_n+b_2\varphi_n+b_3h_n+b_4\lambda_n\varphi_n+\ldots+b_{17}\lambda_n^3+b_{18}\varphi_n^3+b_{19}h_n^3} \\ J_n = B_0 + J_n \cdot B_1 + I_n \cdot B_2 + \frac{P_3(\varphi_n,\lambda_n,h_n)}{P_4(\varphi_n,\lambda_n,h_n)} = \\ = B_0 + J_n \cdot B_1 + I_n \cdot B_2 + \frac{c_0+c_1\lambda_n+c_2\varphi_n+c_3h_n+c_4\lambda_n\varphi_n+\ldots+c_{17}\lambda_n^3+c_{18}\varphi_n^3+c_{19}h_n^3}{1+d_1\lambda_n+d_2\varphi_n+d_3h_n+d_4\lambda_n\varphi_n+\ldots+d_{17}\lambda_n^3+d_{18}\varphi_n^3+d_{19}h_n^3} \end{array} \tag{2.5}$$

The functional model ($\boldsymbol{y}=\mathbf{R}\boldsymbol{x}+\boldsymbol{l}$) for the estimation of affine parameters is

the following:

R is the **design matrix**

$$\mathbf{R} = \left| \begin{array}{c} \mathbf{R}^1 \\ \mathbf{R}^2 \end{array} \right| \begin{array}{ll} \rightarrow & \text{first image} \\ \rightarrow & \text{second image} \end{array} \qquad \mathbf{R}^i = \left| \begin{array}{c} \mathbf{R}_1^i \\ \mathbf{R}_2^i \\ \vdots \\ \mathbf{R}_k^i \end{array} \right| \begin{array}{l} \rightarrow \ 1°\text{GCP} \\ \rightarrow \ 2°\text{GCP} \\ \\ \rightarrow \ k°\text{GCP} \end{array} \tag{2.26}$$

$$\text{for i}^{\text{th}}\text{ GCP} \rightarrow \left\{ \begin{array}{l} \mathbf{R}_j^1 = \left| \begin{array}{ccccccccc} 1 & I_n & J_n & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & I_n & J_n & 0 & \cdots & 0 \end{array} \right| \\ \mathbf{R}_j^2 = \left| \begin{array}{ccccccccc} 0 & \cdots & 0 & 1 & I_n & J_n & 0 & 0 & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 1 & I_n & J_n \end{array} \right| \end{array} \right.$$

$\boldsymbol{x}$ is the **unknown vector**

$$\boldsymbol{x}^T = \left| \begin{array}{cccccccccccc} A_0^1 & A_1^1 & A_2^1 & B_0^1 & B_1^1 & B_2^1 & A_0^2 & A_1^2 & A_2^2 & B_0^2 & B_1^2 & B_2^2 \end{array} \right| \tag{2.27}$$

$\boldsymbol{l}$ is the **known vector**

$$\boldsymbol{l} = \left| \begin{array}{c} \boldsymbol{l}^1 \\ \boldsymbol{l}^2 \end{array} \right| \begin{array}{ll} \rightarrow & \text{first image} \\ \rightarrow & \text{second image} \end{array}$$

$$\boldsymbol{l}^i = \left| \begin{array}{c} \boldsymbol{l}_1^i \\ \boldsymbol{l}_2^i \\ \vdots \\ \boldsymbol{l}_k^i \end{array} \right| \begin{array}{l} \rightarrow \ 1°\text{GCP} \\ \rightarrow \ 2°\text{GCP} \\ \\ \rightarrow \ k°\text{GCP} \end{array} \qquad \boldsymbol{l}_j^i = \left| \begin{array}{c} \left(L_4^i - I^i\right) \\ \left(L_8^i - J^i\right) \end{array} \right| \tag{2.28}$$

$\boldsymbol{y}$ is the **observation vector**

$$\boldsymbol{y} = \left| \begin{array}{c} \boldsymbol{y}^1 \\ \boldsymbol{y}^2 \end{array} \right| \begin{array}{ll} \rightarrow & \text{first image} \\ \rightarrow & \text{second image} \end{array} \qquad \boldsymbol{y}^i = \left| \begin{array}{c} I \\ J \\ I \\ J \\ \vdots \\ I \\ J \end{array} \right| \begin{array}{l} \\ \rightarrow 1°\text{GCP} \\ \\ \rightarrow 2°\text{GCP} \\ \\ \\ \rightarrow k°\text{GCP} \end{array} \tag{2.29}$$

where the superscripts 1 or 2 are related to the first or the second image respectively; A_i, B_i are the affine parameters, (I, J) are the image coordinates and P_j is correlated to equation (2.2). The stochastic model is the same of DLT parameters estimation.

Then, the refined RPCs model (2.5) is applied to each TP.

$$
\begin{aligned}
&I_n = A_0 + I_n \cdot A_1 + J_n \cdot A_2 + \frac{P_1(\varphi_n,\lambda_n,h_n)}{P_2(\varphi_n,\lambda_n,h_n)} = \\
&= A_0 + I_n \cdot A_1 + J_n \cdot A_2 + \frac{a_0+a_1\lambda_n+a_2\varphi_n+a_3h_n+a_4\lambda_n\varphi_n+\ldots+a_{17}\lambda_n^3+a_{18}\varphi_n^3+a_{19}h_n^3}{1+b_1\lambda_n+b_2\varphi_n+b_3h_n+b_4\lambda_n\varphi_n+\ldots+b_{17}\lambda_n^3+b_{18}\varphi_n^3+b_{19}h_n^3} \\
&J_n = B_0 + J_n \cdot B_1 + I_n \cdot B_2 + \frac{P_3(\varphi_n,\lambda_n,h_n)}{P_4(\varphi_n,\lambda_n,h_n)} = \\
&= B_0 + J_n \cdot B_1 + I_n \cdot B_2 + \frac{c_0+c_1\lambda_n+c_2\varphi_n+c_3h_n+c_4\lambda_n\varphi_n+\ldots+c_{17}\lambda_n^3+c_{18}\varphi_n^3+c_{19}h_n^3}{1+d_1\lambda_n+d_2\varphi_n+d_3h_n+d_4\lambda_n\varphi_n+\ldots+d_{17}\lambda_n^3+d_{18}\varphi_n^3+d_{19}h_n^3}
\end{aligned}
\tag{2.5}
$$

The functional model for the TP ground coordinates estimation reads:

$\boldsymbol{y}=\mathbf{A}\boldsymbol{x}+\boldsymbol{l}$

where $\mathbf{A}$ is the **design matrix** (the superscripts 1, 2 indicate the first and second images respectively)

$$
\mathbf{A} = \begin{vmatrix}
\partial\left(I_n^1\right)/\partial\lambda & \partial\left(I_n^1\right)/\partial\varphi & \partial\left(I_n^1\right)/\partial h \\
\partial\left(J_n^1\right)/\partial\lambda & \partial\left(J_n^1\right)/\partial\varphi & \partial\left(J_n^1\right)/\partial h \\
\partial\left(I_n^2\right)/\partial\lambda & \partial\left(I_n^2\right)/\partial\varphi & \partial\left(I_n^2\right)/\partial h \\
\partial\left(J_n^2\right)/\partial\lambda & \partial\left(J_n^2\right)/\partial\varphi & \partial\left(J_n^2\right)/\partial h
\end{vmatrix}
\tag{2.30}
$$

$\boldsymbol{x}$,$\boldsymbol{l}$ and $\boldsymbol{y}$ are the **unknown**, the **known** and the **observation vectors**

$$
\boldsymbol{x}^T = \begin{vmatrix} \varphi & \lambda & h \end{vmatrix} \quad \boldsymbol{l} = \begin{vmatrix}
P_1^1(\varphi_n,\lambda_n,h_n)/P_2^1(\varphi_n,\lambda_n,h_n) \\
P_3(\varphi_n,\lambda_n,h_n)/P_4^1(\varphi_n,\lambda_n,h_n) \\
P_1^2(\varphi_n,\lambda_n,h_n)/P_2^2(\varphi_n,\lambda_n,h_n) \\
P_3^2(\varphi_n,\lambda_n,h_n)/P_4^2(\varphi_n,\lambda_n,h_n)
\end{vmatrix} \quad \boldsymbol{y} = \begin{vmatrix} I^1 \\ J^1 \\ I^2 \\ J^2 \end{vmatrix}
\tag{2.31}
$$

Chapter 3

Model parameters estimation

3.1 Linear Least Squares

The least square method - a very popular technique - is widely used to estimate the numerical values of the parameters, to fit a function and to characterize the statistical properties of estimates.

The main features of the least squares principle are: independence from probability distribution of the observations, capability to supply estimates of minimum variance in the frame of linear and correct estimates, exact calculation of estimates with their precision and possibility to calculate a priori the precision of estimates starting from the one of observation (simulation); on the contrary, such estimation principle is not robust [14], [29].

The general form of an estimation problem, for a linear model, can be written as follows:

- $\boldsymbol{Y_0}$ is the vector of observations (n elements) sampled from the random variable n dimensions $\boldsymbol{Y}$ (observable) defined in the space $\Re^n$. The covariance of $\boldsymbol{Y}$ is known except from a scale factor (variance of unit weight) σ_0^2: $\mathbf{C}_{YY} = \sigma_0^2\mathbf{Q}$ (stochastic model)
- $E(\boldsymbol{Y})$=$\boldsymbol{y}$ is the unknown average of $\boldsymbol{Y}$ and $\boldsymbol{y}\in V^m$ (functional model), where V^m is a linear subspace of the space $\Re^n$, with $m < n$ dimensions

Considering the stochastic and functional models we want to estimate:

- $\hat{\boldsymbol{Y}}$, that is the estimate of the average $\boldsymbol{y}$, under the condition $\hat{\boldsymbol{Y}} \in V^m$
- $\mathbf{C}_{\hat{Y}\hat{Y}}$, that is the estimate of the covariance $\mathbf{C}_{YY}$ (in other words, since $\mathbf{Q}$ is known, we want to estimate $\hat{\sigma}_0^2$ because $\mathbf{C}_{\hat{Y}\hat{Y}} = \hat{\sigma}_0^2 \mathbf{Q}$)

The estimations conditions are the following:

- minimum distance from $\boldsymbol{Y_0}$ according to the matrix $\mathbf{Q}^{-1}$ (least square principle)

$$\left(\boldsymbol{Y_0} - \hat{\boldsymbol{Y}}\right)^T \mathbf{Q}^{-1} \left(\boldsymbol{Y_0} - \hat{\boldsymbol{Y}}\right) = min \tag{3.1}$$

- correctness of the estimate of the variance of unit weight

$$\hat{\sigma}_0^2 = c \left(\boldsymbol{Y_0} - \hat{\boldsymbol{Y}}\right)^T \mathbf{Q}^{-1} \left(\boldsymbol{Y_0} - \hat{\boldsymbol{Y}}\right) \tag{3.2}$$

where c is selected so that $\hat{\sigma}_0^2$ is a correct estimator of σ_0^2 ($E\left(\hat{\sigma}_0^2/\sigma_0^2\right) = \sigma_0^2$)

First of all, we want to list the parameters and the observable estimators needed for the solution of the least squares problem:

- observations: $\boldsymbol{Y_0}(n)$
- unknown parameters: $\boldsymbol{x}(m)$
- functional models: $\boldsymbol{Y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{l}$; where $\mathbf{A}(n,m)$ is the coefficient matrix of unknown parameters (design matrix) and $\boldsymbol{l}(n)$ is the known vector
- stochastic model: $\mathbf{C}_{YY} = \sigma_0^2 \mathbf{Q}$, where $\mathbf{Q}(n,n)$ is the cofactor matrix of the observations

The parameter estimators are:

$$\begin{cases} \hat{\boldsymbol{x}} = \left(\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\right)^{-1} \mathbf{A}^T\mathbf{Q}^{-1}(\boldsymbol{Y_0} - \boldsymbol{l}) & \\ \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} = \mathbf{N}(m,m) & \text{normal matrix} \\ \mathbf{A}^T\mathbf{Q}^{-1}(\boldsymbol{Y_0} - \boldsymbol{l}) = \boldsymbol{t}(m) & \text{normal known vector} \end{cases} \tag{3.3}$$

The observable and residual estimators are:

$$\begin{cases} \hat{\boldsymbol{Y}} = \mathbf{A}\hat{\boldsymbol{x}} + \boldsymbol{l} = \mathbf{A}\left(\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\right)^{-1} \mathbf{A}^T\mathbf{Q}^{-1}(\boldsymbol{Y_0} - \boldsymbol{l}) + \boldsymbol{l} \\ \hat{\boldsymbol{U}} = \boldsymbol{Y_0} - \hat{\boldsymbol{Y}} = \left[\mathbf{I} - \mathbf{A}\left(\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\right)^{-1} \mathbf{A}^T\mathbf{Q}^{-1}\right](\boldsymbol{Y_0} - \boldsymbol{l}) \end{cases} \tag{3.4}$$

The estimator $\hat{\sigma}_0^2$ is:

$$\hat{\sigma}_0^2 = \frac{\left(\boldsymbol{Y_0} - \hat{\boldsymbol{Y}}\right)^T \mathbf{Q}^{-1} \left(\boldsymbol{Y_0} - \hat{\boldsymbol{Y}}\right)}{n - m} \tag{3.5}$$

The covariance estimators are:

$$\begin{cases} \mathbf{C}_{\hat{x}\hat{x}} = \hat{\sigma}_0^2 \left(\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\right)^{-1} & \text{covariance of parameters} \\ \mathbf{C}_{\hat{Y}\hat{Y}} = \hat{\sigma}_0^2 \mathbf{A} \left(\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\right)^{-1} \mathbf{A}^T & \text{covariance of estimated observations} \\ \mathbf{C}_{\hat{U}\hat{U}} = \hat{\sigma}_0^2 \left[\mathbf{Q} - \mathbf{A} \left(\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}\right)^{-1} \mathbf{A}^T\right] & \text{covariance of residuals} \end{cases} \tag{3.6}$$

3.2 Nonlinear Least Squares

In the least squares method, the nonlinear problems have some drawbacks: the solution is not optimum and unique. The general nonlinear functional model reads:

$$f(x, y) = 0 \qquad \boldsymbol{x} \in \Re^n,\ \boldsymbol{y} \in \Re^n \tag{3.7}$$

and we search $\hat{\boldsymbol{y}}$ e $\hat{\boldsymbol{x}}$ such as

$$\begin{cases} \left(\boldsymbol{Y_0} - \hat{\boldsymbol{y}}\right)^T \mathbf{Q}^{-1} \left(\boldsymbol{Y_0} - \hat{\boldsymbol{y}}\right) = min \\ f(\hat{y}, \hat{x}) = 0 \end{cases} \tag{3.8}$$

If we know the approximate values $\tilde{\boldsymbol{y}}$, $\tilde{\boldsymbol{x}}$ and supposing that $f(x, y)$ can be well approximate with a linear function, we can write:

$$\begin{cases} \hat{y} = \tilde{y} + \hat{\eta} \\ \hat{x} = \tilde{x} + \hat{\xi} \end{cases} \tag{3.9}$$

The equation (3.7) can be linearized in the following way:

$$f(\tilde{y}, \tilde{x}) + \frac{\partial \tilde{f}}{\partial y}\hat{\eta} + \frac{\partial \tilde{f}}{\partial x}\hat{\xi} = 0 \tag{3.10}$$

The equation (3.10) is the new "linear" model and the new variables are ξ (parameters) and η (observable).

Using a more concise terminology, we can write:

$$\begin{cases} \frac{\partial \tilde{f}}{\partial x} = \mathbf{A} & \text{A is the design matrix} \\ -\frac{\partial \tilde{f}}{\partial y} = \mathbf{D} & \text{D is the observation coefficient matrix} \\ f(\tilde{y}, \tilde{x}) = \boldsymbol{l} & \text{l is the known vector} \\ \eta_0 = \boldsymbol{Y_0} - \tilde{y} & \\ \mathbf{C}_{\eta\eta} = \mathbf{C}_{YY} = \sigma_0^2 \mathbf{Q} & \end{cases} \tag{3.11}$$

Considering (3.11), the equation (3.10) becomes:

$$\mathbf{D}\eta = \mathbf{A}\xi + \boldsymbol{l} \tag{3.12}$$

Therefore the constrained minimum problem reads:

$$\begin{cases} (\eta_0 - \hat{\eta})^T \mathbf{Q}^{-1} (\eta_o - \hat{\eta}) = \min \\ \mathbf{D}\hat{\eta} = \mathbf{A}\hat{\xi} + \boldsymbol{l} \end{cases} \tag{3.13}$$

and the estimators $\hat{\eta}$, $\hat{\xi}$ are

$$\begin{cases} \hat{\eta} = \eta_0 - \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U_0} \\ \hat{\xi} = \mathbf{N}^{-1}\mathbf{A}^T\mathbf{K}^{-1}(\mathbf{D}\eta_0 - \boldsymbol{l}) \end{cases} \tag{3.14}$$

where

$$\begin{cases} \mathbf{K} = \mathbf{D}\mathbf{Q}\mathbf{D}^T & \\ \mathbf{N} = \mathbf{A}^T\mathbf{K}^{-1}\mathbf{A} & \text{normal matrix} \\ \boldsymbol{U_0} = \mathbf{D}\eta_0 - \mathbf{A}\xi - \boldsymbol{l} & \text{residual vector} \end{cases} \tag{3.15}$$

If the approximate values $\tilde{y}$, $\tilde{x}$ are very raw, that is the linear approximation is not adequate, the estimate procedure is iterated; for each step the approximate values are equal to the estimated value of the previous iteration.

The iterative algorithm is:

- step 1

$$\hat{y}_0 = \tilde{y} \qquad \hat{x}_0 = \tilde{x} \tag{3.16}$$

- step $i+1$

$$\begin{aligned} &\tilde{y}_{i+1} = \hat{y}_i \qquad \tilde{x}_{i+1} = \hat{x}_i \\ &\hat{y}_{i+1} = \hat{y}_i + \hat{\eta}_{i+1} \qquad \hat{x}_{i+1} = \hat{x}_i + \hat{\xi}_{i+1} \\ &\mathbf{D}_{i+1} = -\left[\tfrac{\partial g(\hat{y}_i, \hat{x}_i)}{\partial y}\right] \qquad \mathbf{A}_{i+1} = \left[\tfrac{\partial g(\hat{y}_i, \hat{x}_i)}{\partial x}\right] \qquad \boldsymbol{l}_{i+1} = g\left(\hat{y}_i, \hat{x}_i\right) \\ &\eta_{0,i+1} = \boldsymbol{Y_0} - \hat{y}_i \qquad \mathbf{C}_{\eta\eta} = \mathbf{C}_{YY} \qquad \mathbf{D}_i\hat{\eta}_{i+1} = \mathbf{A}_i\hat{\xi}_{i+1} + \boldsymbol{l}_i \end{aligned} \tag{3.17}$$

the algorithm stops when the estimated variance of unit weight reaches a minimum.

Corrections of observations

The third equation in (3.17) shows that the estimated value of the observations, $\hat{y}$, derives from the addition between the approximate value and its estimated correction

$$\hat{y}_{i+1} = \tilde{y}_{i+1} + \hat{\eta}_{i+1} = \hat{y}_i + \hat{\eta}_{i+1} \tag{3.18}$$

The iterative procedure can be schematized in the following way:

- step $0 \Rightarrow i = 0$: $\tilde{y_0} = \boldsymbol{Y_0}$ where $\boldsymbol{Y_0}$ are the observations (i, j, X, Y, Z)
- step $1 \Rightarrow i = 1$: $\hat{y_1} = \hat{y}_0 + \hat{\eta}_1 = Y_0 + \hat{\eta}_1$
- step $2 \Rightarrow i = 2$: $\hat{y_2} = \hat{y}_1 + \hat{\eta}_2 = Y_0 + \hat{\eta}_1 + \hat{\eta}_2$
- $\cdots$

Considering the equations (3.11) and (3.14) the estimated value $\hat{\eta}$ can be calculated:

$$\begin{cases} \hat{\eta}_{0,i} = \boldsymbol{Y_0} - \tilde{y}_i = \boldsymbol{Y_0} - \hat{y}_{i-1} \\ \hat{\eta}_i = \hat{\eta}_{0,i} - \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,i} \end{cases} \tag{3.19}$$

- step $1 \Rightarrow i = 1$:

$$\begin{cases} \hat{\eta}_{0,1} = \boldsymbol{Y_0} - \hat{y}_0 = \boldsymbol{Y_0} - \boldsymbol{Y_0} = 0 \\ \hat{\eta}_1 = -\mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,1} \end{cases} \tag{3.20}$$

- step $2 \Rightarrow i = 2$:

$$\begin{cases} \hat{\eta}_{0,2} = \boldsymbol{Y_0} - \hat{y}_1 = \boldsymbol{Y_0} - (\boldsymbol{Y_0} + \hat{\eta}_1) = -\hat{\eta}_1 = \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,1} \\ \hat{\eta}_2 = \hat{\eta}_{0,2} - \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,2} = \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,1} - \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,2} \end{cases} \tag{3.21}$$

- step $3 \Rightarrow i = 3$:

$$\begin{cases} \hat{\eta}_{0,3} = \boldsymbol{Y_0} - \hat{y}_2 = \boldsymbol{Y_0} - (\boldsymbol{Y_0} + \hat{\eta}_1 + \hat{\eta}_2) = -\hat{\eta}_1 - \hat{\eta}_2 = \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,2} \\ \hat{\eta}_3 = \hat{\eta}_{0,3} - \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,3} = \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,2} - \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,3} \end{cases} \tag{3.22}$$

- $\cdots$
- step i^{th}:

$$\begin{cases} \hat{\eta}_{0,i} = \boldsymbol{Y_0} - \hat{y}_{i-1} = -\sum_{k=1}^{i-1} \hat{\eta}_k \\ \hat{\eta}_i = \hat{\eta}_{0,i} - \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,i} = \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,i-1} - \mathbf{Q}\mathbf{D}^T\mathbf{K}^{-1}\boldsymbol{U}_{0,i} \end{cases} \tag{3.23}$$

3.3 Cholesky decomposition

Cholesky decomposition is a special, very efficient, triangular decomposition for a square, symmetric and positive definite matrix [30], [28].

A generic square matrix $\mathbf{C}_{n\times n}$ is symmetric if

$$c_{ij} = c_{ji} \qquad for\ i,j = 1,\cdots,n \tag{3.24}$$

and it is definite positive if

$$v \cdot \mathbf{C} \cdot v > 0 \qquad for\ all\ vectors\ v \tag{3.25}$$

in other words a matrix $\mathbf{C}$ is positive definite if all its eigenvalues are all positive.

Cholesky decomposition constructs an upper triangular matrix $\mathbf{T}$ such as:

$$\mathbf{T}^t\mathbf{T} = \mathbf{C} \tag{3.26}$$

that is

$$\begin{array}{ll} c_{11} = t_{11}t_{11} & \\ c_{1j} = t_{11}t_{1j} & (j>1) \\ c_{ii} = t_{ii}t_{ii} + \sum_{k=1}^{i-1} t_{ki}t_{ki} & (i \neq j) \\ c_{ij} = t_{ii}t_{ij} + \sum_{k=1}^{i-1} t_{ki}t_{kj} & (j>i) \end{array} \tag{3.27}$$

The expressions used for the computation of $\mathbf{T}$ elements are taken from the previous considerations (3.27)

$$\begin{array}{ll} t_{11} = \sqrt{c_{11}} & \\ t_{1j} = c_{1j}/t_{11} & (j>1) \\ t_{ii} = \sqrt{c_{ii} - \sum_{k=1}^{i-1} t_{ki}t_{ki}} & (i \neq j) \\ t_{ij} = (c_{ij} - \sum_{k=1}^{i-1} t_{ki}t_{kj})/t_{ii} & (j>i) \end{array} \tag{3.28}$$

3.3.1 Cholesky decomposition for system solution

Cholesky decomposition can be used to solve the generic system with n equation and n unknown parameters

$$\mathbf{C}\boldsymbol{x} + \boldsymbol{d} = 0 \tag{3.29}$$

Using Cholesky decomposition ($\mathbf{C} = \mathbf{T}^t\mathbf{T}$), we can write

$$\mathbf{T}^t\mathbf{T}\boldsymbol{x} = -\boldsymbol{d} \tag{3.30}$$

$$\mathbf{T}^t\boldsymbol{y} = \boldsymbol{d} \tag{3.31}$$

$$\mathbf{T}\boldsymbol{x} = -\boldsymbol{y} \tag{3.32}$$

The solution of the two systems (3.31) and (3.32) supplies the solution of (3.30), that is our starting system.

The following equations allow the computation of $\boldsymbol{y}$ and $\boldsymbol{x}$ vectors

$$\begin{aligned} y_1 &= d_1/t_{11} \\ y_i &= (d_i - \sum_{k=1}^{i-1} t_{ki} y_k)/t_{ii} \qquad (i \neq 1) \end{aligned}$$

$$\begin{aligned} x_n &= -y_n/t_{nn} \\ x_i &= -(y_i + \sum_{k=i+1}^{n} t_{ik} x_k)/t_{ii} \qquad (i \neq n) \end{aligned}$$

3.3.2 Cholesky decomposition for matrix inversion

The inversion of the matrix $\mathbf{C}$ can be computed with Cholesky decomposition

$$\begin{aligned} \mathbf{C}^{-1} = (\mathbf{T}^t\mathbf{T})^{-1} &= \mathbf{T}^{-1}(\mathbf{T}^t)^{-1} \\ \mathbf{T}\mathbf{C}^{-1} &= (\mathbf{T}^t)^{-1} \end{aligned} \tag{3.33}$$

Given that the inverse of a matrix is symmetric, the computation of $\mathbf{C}^{-1}$ can be calculated working only in the upper triangle

$$\begin{aligned} &\gamma_{nn} = 1/t_{nn}^2 \\ &\gamma_{ij} = -\textstyle\sum_{k=i+1}^{n} t_{ik}\gamma_{kj}/t_{ii} \qquad (j > i \, e \, j \geq k) \\ &\gamma_{ij} = -\textstyle\sum_{k=i+1}^{n} t_{ik}\gamma_{jk}/t_{ii} \qquad (j > i \, e \, j \leq k) \\ &\gamma_{ii} = (1/t_{ii} - \textstyle\sum_{k=i+1}^{n} t_{ik}\gamma_{ik})/t_{ii} \qquad (j \neq i) \end{aligned} \tag{3.34}$$

where γ_{ij} are the elements of $\mathbf{C}^{-1}$ matrix.

3.4 Singular Value Decomposition

Sometimes the design matrix is likely to be close to singularity, so that rank estimation and estimable parameter selection is mandatory. In this respect SVD and QR decomposition are quite useful tools and they will be shortly recalled here. The Singular Value Decomposition (SVD) and the QR decomposition are employed to solve the linearized collinearity equations system in the LS sense

[26], [27]. As usual, the solution is obtained iteratively due to non-linearity of the system; the iterative procedure stops when the estimated variance of the unit weight observation stabilizes.

The Singular Value Decomposition (SVD) is a very powerful technique to deal with sets of equations or matrices that are either singular or numerically very close to being singular. The SVD of a matrix $\mathbf{A} \in \Re^{m \cdot n}$ (with $m \geq n$) is any factorization of the form:

$$\mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^T \tag{3.35}$$

where $bfW \in \Re^{n \cdot n}$ is a diagonal matrix with positive or zero elements (w_{ij}) that are the singular values of $\mathbf{A}$; $\mathbf{U} \in \Re^{m \cdot n}$ and $\mathbf{V} \in \Re^{m \cdot n}$ are orthogonal matrices, whose columns (u_j, v_j) are called the left and right singular vectors. For a system of linear equations $\mathbf{A}\boldsymbol{x}$=$\boldsymbol{b}$, using the SVD we can write ([27]):

$$\mathbf{U}\mathbf{W}\mathbf{V}^T\boldsymbol{x} = \boldsymbol{b} \tag{3.36}$$

and the LS solution $\boldsymbol{x}$ minimizes $||\mathbf{A}\boldsymbol{x}-\boldsymbol{b}||_2^2$. Since the orthogonal matrix preserves the norm, for any $\boldsymbol{x} \in \Re^n$ we have:

$$\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \left\|(\mathbf{U}^T\mathbf{A}\mathbf{V})(\mathbf{V}^T\boldsymbol{x}) - \mathbf{U}^T\boldsymbol{b}\right\|_2^2 = \sum\nolimits_{i=1}^{r}(w_i z_i - u_i^T\boldsymbol{b})^2 + \sum\nolimits_{i=r+1}^{m}(u_i^T\boldsymbol{b})^2 \tag{3.37}$$

where $z = V^T x$ and r is the rank of A. $\|Ax - b\|_2{}^2 = \min$ holds, if

$$\sum_{i=1}^{r}(w_i z_i - u_i^T\boldsymbol{b})^2 = 0 \tag{3.38}$$

then using the SVD, the LS problem is now in form of a diagonal matrix, and fi-nally

$$x_i = \begin{cases} x_i = \frac{u_i^T\boldsymbol{b}}{w_i}v_i & \text{if } \mathrm{w_i} \neq 0 \\ \text{undetermined} & \text{if } \mathrm{w_i} = 0 \end{cases} \tag{3.39}$$

The advantage of using the SVD is that it can reliably handle the rank deficient case as well as the full rank case.

3.5 QR Decomposition

The QR decomposition of a matrix $\mathbf{A} \in \Re^{m \cdot n}$ (with $m \geq n$) is given by:

$$\mathbf{A} = \mathbf{Q}\mathbf{R} \tag{3.40}$$

where $\mathbf{Q} \in \Re^{m \cdot m}$ is an orthogonal matrix and $\mathbf{R} \in \Re^{m \cdot n}$ is an upper triangular matrix. If the rank of $\mathbf{A}$ is equal to n, the first n columns of $\mathbf{Q}$ form an orthonormal basis for the $Rank(\mathbf{A})$. Thus, the calculation of the $\mathbf{QR}$ factorization is a way to compute an orthonormal basis for a set of vectors. The standard algorithm for the QR decomposition involves sequential evaluation of Householder transformations. An appropriate Householder matrix, applied to a given matrix, can zero all the elements, situated below a given element, in a column of the matrix. For the first column of the matrix $\mathbf{A}$, an appropriate matrix $\mathbf{H}_1$ is evaluated, which puts on zero all the elements below the first element in the first column of $\mathbf{A}$. Similarly $\mathbf{H}_2$ zeroes all elements in the second column below the second element and so on up to $\mathbf{H}_{n-1}$

$$\mathbf{R} = \mathbf{H}_{n-1} \cdots \mathbf{H}_1 \mathbf{A} \tag{3.41}$$

where $\mathbf{Q}^T = \mathbf{H}_{n-1} \ldots \mathbf{H}_1$, i.e., $\mathbf{Q} = \mathbf{H}_1 \ldots \mathbf{H}_{n-1}$. The generic matrix $\mathbf{H}_i$ zeroes all elements in the first column below the first element for a sub-matrix of $\mathbf{A}$ ($\mathbf{A}_i \in \Re^{((m-i)\cdot(n-i))}$). If $\mathbf{A}$ is rank deficient, the $\mathbf{QR}$ factorization does not give a basis for the $Rank(\mathbf{A})$. In this case to calculate an orthonormal basis for $Rank(\mathbf{A})$, it is necessary to compute the $\mathbf{QR}$ decomposition of a column-permuted version of $\mathbf{A}$, i.e., $\mathbf{AP} = \mathbf{QR}$ [27]

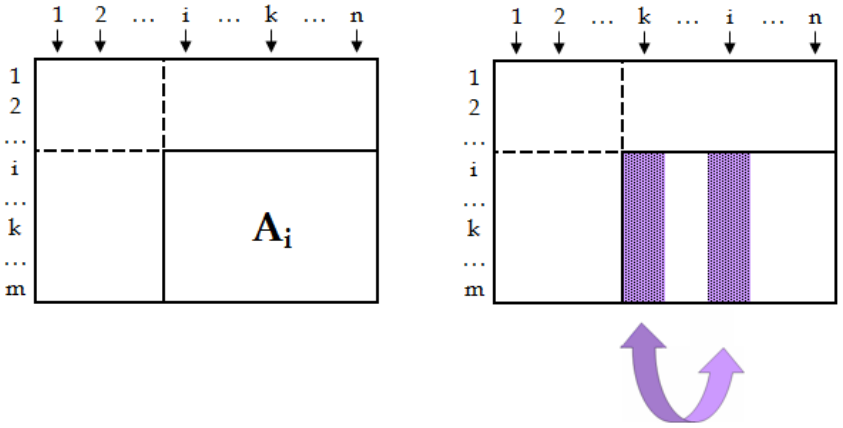
$$\mathbf{Q}^T\mathbf{AP} = \begin{array}{c} \left[\begin{array}{cc} \mathbf{R}_{11} & \mathbf{R}_{12} \\ 0 & 0 \end{array}\right] \begin{array}{l} \rightarrow \\ \rightarrow \end{array} \begin{array}{l} r \\ m-r \end{array} \\ \begin{array}{cc} \downarrow & \downarrow \\ r & n-r \end{array} \end{array} \tag{3.42}$$

where $\mathbf{P}$ is a permutation, r is the rank of $\mathbf{A}$, $\mathbf{R}_{11}$ is an upper triangular and non singular matrix and $\mathbf{Q}$ and $\mathbf{P}$ are products of Householder matrices $\mathbf{Q} = \mathbf{H}_1 \ldots \mathbf{H}_r$, $\mathbf{P} = \mathbf{P}_1 \ldots \mathbf{P}_r$. For understanding the role of the permutation matrix, it is necessary to define the vector $\boldsymbol{N} \in \Re^m$ for a generic matrix : $\mathbf{A} \in \Re^{m \cdot n}$

$$\mathbf{A}_{mn} = \left| \begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \cdots & & \vdots \\ a_{m1} & & \cdots & a_{mn} \end{array} \right| \tag{3.43}$$

$$\boldsymbol{N}_m = \left| \begin{array}{ccc} \sum_{k=1}^{m} a_{k1}^2 & \sum_{k=1}^{m} a_{k2}^2 & \cdots \quad \sum_{k=1}^{m} a_{kn}^2 \end{array} \right| \tag{3.44}$$

The element of the $\boldsymbol{N}$ are the square value of norm calculated for each column of $\mathbf{A}$. The permutation matrix $\mathbf{P}$ applied at the generic matrix $\mathbf{A}$ makes a matrix $\mathbf{A}_P = \mathbf{AP}$ such that the elements of the corresponding vector $\boldsymbol{N}$ are placed in



(a) A_i before the permutation (b) A_i after the permutation

Figure 3.1: A_i before and after the permutation

descending order. As for the generic matrix $\mathbf{H}_i$, the generic matrix $\mathbf{P}_i$ permutes the column of a sub-matrix of $\mathbf{A}_i \in \Re^{((m-i)\cdot(n-i))}$; if k is the column with the maximum value of norm, the permutation matrix $\mathbf{P}_i$ exchanges the columns i and k.

In a system of linear equations $\mathbf{A}\boldsymbol{x}=\boldsymbol{b}$, if $\mathbf{A} \in \Re^{m\cdot n}$ and has a rank r, the $\mathbf{QR}$ decomposition produces the factorization $\mathbf{AP} = \mathbf{QR}$ where $\mathbf{R}$ is described in the equation (5.1). As for the LS problem we have:

$$\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \left\|(\mathbf{Q}^T\mathbf{AP})(\mathbf{P}^T\boldsymbol{x}) - \mathbf{Q}^T\boldsymbol{b}\right\|_2^2 = \|\mathbf{R}_{11}\boldsymbol{t} - (\boldsymbol{c} - \mathbf{R}_{12}\boldsymbol{z})\|_2^2 + \|\boldsymbol{d}\|_2^2 \tag{3.45}$$

where

$$\begin{aligned} \mathbf{P}^T\boldsymbol{x} &= \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{z} \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} r \\ n-r \end{matrix} \\ \mathbf{Q}^T\boldsymbol{b} &= \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \end{matrix} \begin{matrix} r \\ m-r \end{matrix} \end{aligned} \tag{3.46}$$

If $\boldsymbol{x}$ is a LS minimizer we have

$$\boldsymbol{x} = \mathbf{P}\begin{bmatrix} \mathbf{R}_{11}^{-1}(\boldsymbol{c} - \mathbf{R}_{12}\boldsymbol{z}) \\ \boldsymbol{z} \end{bmatrix} \tag{3.47}$$

If $\boldsymbol{z}$ is a set of zeroes in this expression, we obtain the basic solution:

$$\boldsymbol{x}_B = \mathbf{P} \begin{bmatrix} \mathbf{R}_{11}^{-1}\boldsymbol{c} \\ 0 \end{bmatrix} \tag{3.48}$$

$\boldsymbol{x}_B$ has at most r non-zero components and so $\mathbf{A}\boldsymbol{x}_B$ involves a subset of $\mathbf{A}$ columns.

3.5.1 Subset Selection Using SVD and QR

For a system of linear equations ($\mathbf{A}\boldsymbol{x}$=$\boldsymbol{b}$), with $\mathbf{A} \in \Re^{m \cdot n}$ (with $m \geq n$) it is necessary to select the estimable parameters. We describe an SVD based subset selection procedure, due to Golub, Klema and Stewart [27], that proceeds as follows:

- we compute the SVD $\mathbf{A} = \mathbf{UWV}^T$ and use it to determine a rank estimate r
- with the $\mathbf{QR}$ decomposition $\mathbf{QR} = \mathbf{AP}$ we select an independent subset of $\mathbf{A}$ columns; if $\mathbf{R}_{11}\boldsymbol{x}_B = \mathbf{Q}^T\ \boldsymbol{b}$ with $\boldsymbol{x}_B \in \Re^r$ and we set

  $$\boldsymbol{t} = \mathbf{P} \begin{bmatrix} \boldsymbol{x}_B \\ 0 \end{bmatrix} \tag{3.49}$$

  then $\mathbf{A}\boldsymbol{t}$ is an approximate LS predictor of $\boldsymbol{b}$ that involves the first r columns of $\mathbf{AP}$. The permutation matrix $\mathbf{P}$ is calculated so that the columns of the matrix $\mathbf{B}_1 \in \Re^{m \cdot r}$ in $\mathbf{AP} = [\mathbf{B}_1, \mathbf{B}_2]$ are "sufficiently independent"
- we predict $\boldsymbol{b}$ with the vector $\mathbf{A}\boldsymbol{t}$ where $\boldsymbol{t}$ is described in the equation (3.49), and $\boldsymbol{z}$ minimizes minimizes $||\mathbf{B}_1\boldsymbol{x}_B - \boldsymbol{b}||_2$

Chapter 4

Application of the Rigorous orientation models

4.1 Introduction and data set

In this chapter they are presented some results obtained using the previously described rigorous orientation models. The orientation models are implemented in a scientific software (SISAR - Software per Immagini Satellitari ad Alta Risoluzione) developed at the DITS, Area di Geodesia e Geomatica, Sapienza Università di Roma, in C++ language.

SISAR results are compared with orientation models implemented in others commercial software (OrthoEngine 10.0 PCI Geomatica, Erdas 9.0 Leica Geosystem) [16], [11], [25]. The available images for the experimentation have been acquired by several sensors. Their features are summarized in Table 4.1 [1]

[1] EROS A Rome (R1):$ITA1 - e1038452$;
EROS A Rome (R2): $ITA1 - e1090453$;
EROS A Rome (R4): $MBT1 - E1009023$;
QuickBird Salerno "joint" is obtained to stitch three QuickBird images in order to have a single "strip" image:
QuickBird Salerno (*P001): $05JUL17100900 - P1BS - 005520834030\_01\_P001$;
QuickBird Salerno (*P002): $05JUL17100903 - P1BS - 005520834030\_01\_P002$;
QuickBird Salerno (*P003): $05JUL17100906 - P1BS - 005520834030\_01\_P003$;
QuickBird Augusta (*P001): $04JAN06093201 - P1BS - 000000130187\_01\_P001$;
QuickBird Augusta (*P002): $04JAN06093307 - P1BS - 000000130187\_01\_P002$;
QuickBird Rome (level 1A): $02JUN03100558 - P1BS - 000000032060\_01\_P001$;

Table 4.1: Data set

Sensor	**Area**	**GSD [m]**	**Off-nadir angle [°] start**	**end**	**Scene coverage [KmxKm]**	**GPs**
EROS A	Rome (R1)	1.8	9.1	9.4	13x10	49
	Rome (R2)	2.6	31.0	40.1	17x12	49
	Rome (R4)	1.9	15.3	16.4	14x10	49
QuickBird	Rome (level 1A)	0.61	3.0		17x17	49
	Rome (level 1B)	0.60	2.2		17x17	24
	Augusta (*P001)	0.77	29.2		21x20	39
	Augusta (*P002)	0.75	28.2		20x19	39
	Salerno (*P001)	0.67	19.0		18x18	17
	Salerno (*P002)	0.67	20.0		18x18	34
	Salerno (*P003)	0.68	20.9		18x18	18
	Salerno "joint"	0.67	20.0		48x19	57
Ikonos	Rome	1.00	18.7		11x10	27
	Bagnoli1	1.00	23.0		13x9	25
	Bagnoli2	1.00	24.3		13x9	25
	Bagnoli3	1.00	23.0		13x9	15
	Bagnoli4	1.00	24.3		13x9	15
Cartosat-1	Rome (bandA)	2.50	4.9		7.5x30	43
	Rome (bandF)	2.50	26.1		7.5x30	43
	CastelGandolfo (bandA)	2.50	12.3		30x30	25
	CastelGandolfo (bandF)	2.50	28.2		30x30	25
	Warsaw (bandA)	2.50	5.0		30x30	29
	Warsaw (bandF)	2.50	26.0		30x30	29
	Mausanne (bandA)	2.50	14.5		30x30	32
	Mausanne (bandF)	2.50	29.1		30x30	32
WorldView-1	Augusta(4505 R1C1)	0.55	18.4		14x19	16
	Augusta (4505 R1C2)	0.55	18.4		12x19	17
	Augusta (4545 R1C1)	0.56	20.3		15x19	16
	Augusta (4545 R1C2)	0.56	20.3		11x19	17
	Rome (level 1B)	0.51	6.7		17x15	24
GeoEye-1	Rome	0.50	12.0		15x17	37

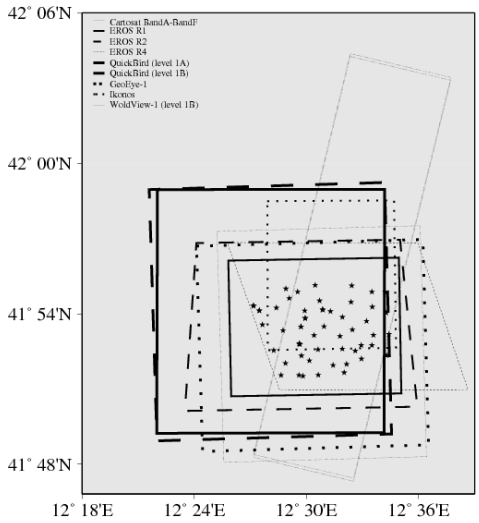


Figure 4.1: Distribution of the GPs in the Rome area

All the images of Rome (3 EROS-A, 2 QuickBird (1 level 1A, 1 level 1B), 1 Ikonos, 1 Cartosat-1 stereopair, 1 GeoEye-1) cover areas of different extensions (Fig.4.1); the GPs were surveyed with static or fast static procedures by a Trimble 5700 GPS receiver and their coordinates were estimated by Trimble Geomatic Office software with respect to the available GPS permanent stations MOSE at Rome Faculty of Engineering. The mean horizontal and vertical accuracies of the coordinates are between 10 and 20 cm, considering the monumentation accuracy of these points, which are already existing man-made objects located on the ground.

The Augusta (Sicily) scenes are 4, 1 stereopair acquired by QuickBird and 1 stereopair acquired by WorldView-1, with ground points (GPs) positioned in

QuickBird Rome (level 1B): $05APR28101432 - P2AS - 005746807010\_01\_P001$;
Ikonos Rome: po_15194;
Ikonos Bagnoli1: $po\_918\_pan\_0000010001$;
Ikonos Bagnoli2: $po\_918\_pan\_0010000001$;
Ikonos Bagnoli3: $po\_918\_pan\_0000010000$;
Ikonos Bagnoli4: $po\_918\_pan\_0010000000$;
WorldView-1 Augusta (4505 R1C1): $08JUN20094505 - P1BS\_R1C1 - 052051590010\_01\_P001$;
WorldView-1 Augusta (4505 R2C1): $08JUN20094505 - P1BS\_R2C1 - 052051590010\_01\_P001$;
WorldView-1 Augusta (4545 R1C1): $08JUN20094545 - P1BS\_R1C1 - 052051590010\_01\_P001$;
WorldView-1 Augusta (4545 R2C1): $08JUN20094545 - P1BS\_R2C1 - 052051590010\_01\_P001$;
WorldView-1 Rome (level 1B): $08FEB15100038 - P2AS - 005733565010\_01\_P001$;
GeoEye-1 Rome: $po\_3800668$;

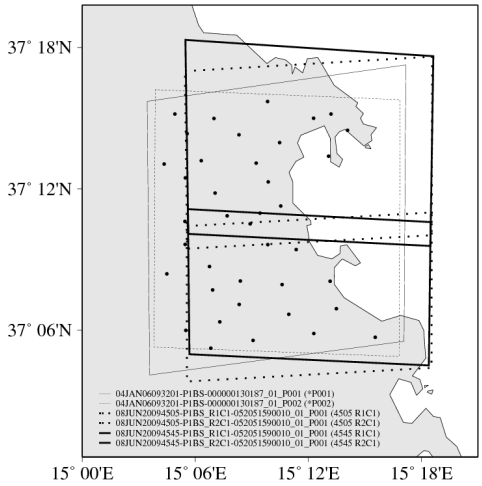


Figure 4.2: Distribution of the GPs in the Augusta area

the overlapping area (Fig.4.2). Each WorldView-1 image is broken up into 2 tiles (4505 R1C1, 4505 R2C1 and 4545 R1C1, 4545 R2C1) due to their large size (exceeding 2 GB) with respect to the available media types.

The Salerno image includes three different QuickBird standard format images coming from the same orbital segment (Fig.4.3); the particularity of this image is the latitude extension (around 48 Km).

The Bagnoli scenes (near Naples) are 2 stereopairs, overlapped approximately for 20 percent.

The Cartosat-1 images are 3 stereopairs, one of CastelGandolfo area (middle Italy), one of Warsaw (Poland) and one of Mausanne (France).

The GPs for the Augusta and Salerno images were surveyed by geodetic quality GPS in RTK mode; the mean horizontal and vertical coordinate accuracies are between 5 and 10 cm.

The GPs distribution on Bagnoli stereopairs is not homogeneous due to the sea in the South-West area (Fig.4.4). The GPs of Bagnoli were acquired by static GPS surveys using Topcon Legacy receivers in post processing procedure with respect to quite far (100 to 200 km) permanent stations due to the lack of available near GPs permanent stations and logistic difficulties to estabilish a master GPs station just for the survey period. The mean horizontal and vertical accuracies of the coordinates are between 10 and 20 cm.

The GPs for the Castelgandolfo area were collected in RTK mode, the mean

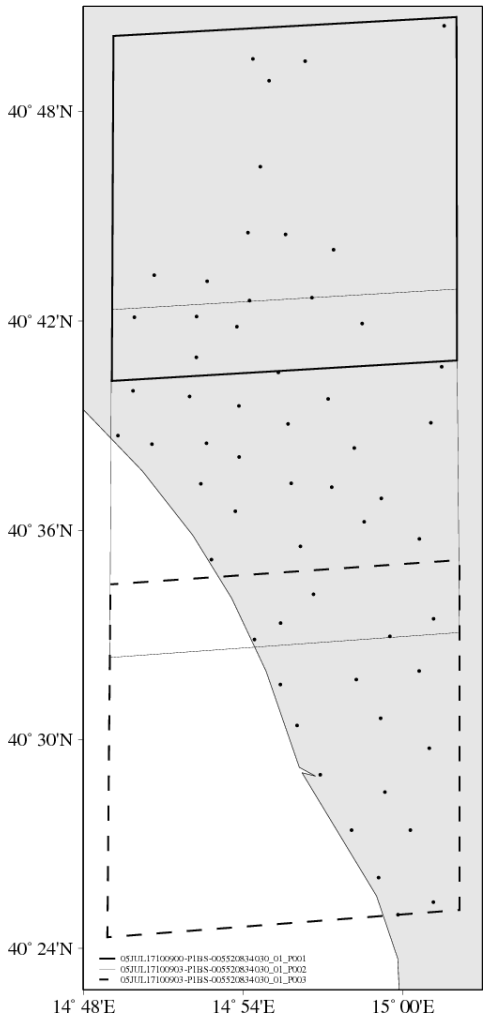


Figure 4.3: Distribution of the GPs in the Salerno area

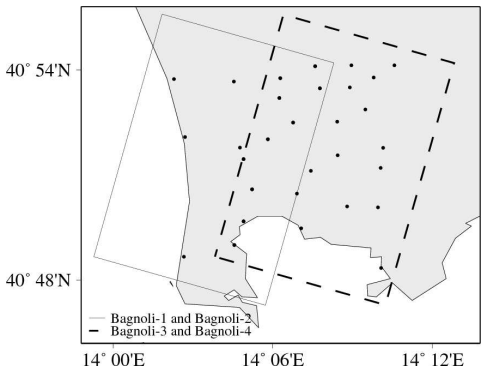


Figure 4.4: Distribution of the GPs in the Bagnoli area

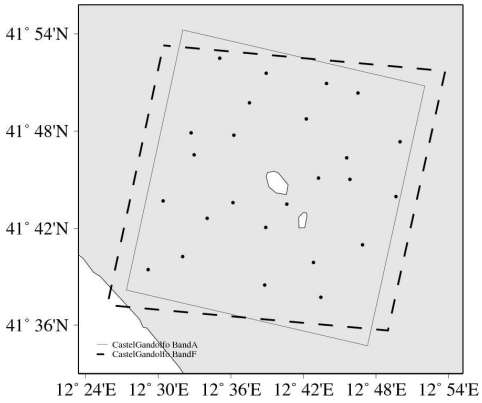


Figure 4.5: Distribution of the GPs in the Castelgandolfo area

horizontal and vertical coordinate accuracies are around 15 cm (Fig.4.5). The GPs for Warsaw and Mausanne were surveyed by people from other research groups and used in others international project (for example Cartosat-1 Scientific Assessment Project managed by ISPRS-ISRO) with GPS receiver; their coordinates mean horizontal and vertical accuracies are between 10 and 20 cm.

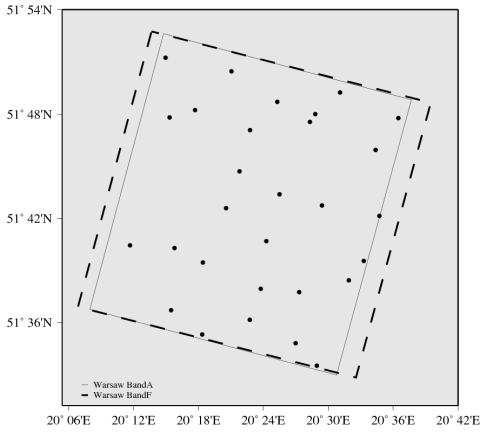


Figure 4.6: Distribution of the GPs in the Warsaw area

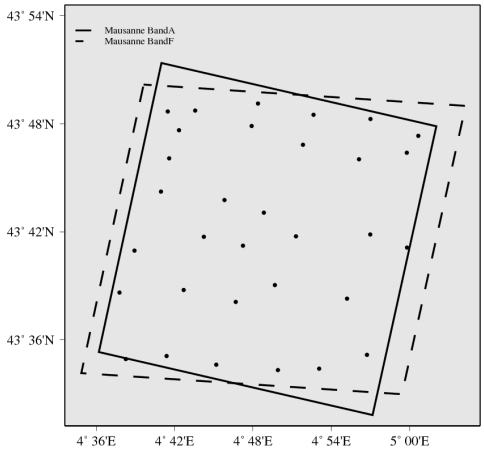


Figure 4.7: Distribution of the GPs in the Mausanne area

The ground coordinates for all the GPs are expressed in the WGS84 system, while the orthometric heights were obtained applying geoid undulations from the ITALGEO95 public model for the National Area, whereas for Mausanne and Warsaw imagery the orthometric heights were supplied by the other research group together the GPs.

For each image and for each software the orientation was carried out several times, varying the number of GCPs, and the related accuracies, represented by the RMSE computed over Check Points (CPs) residuals (RMSE CP), were computed and analyzed. The RMSEs were computed both for the North and East residual components separately. This method is known like Hold-Out Validation (HOV). This method has the advantage of being simple and easy to compute, but it also has some drawbacks, as it is generally not reliable and it is not applicable when a low number of ground points is available. First of all, once the two sets are selected, accuracy estimate is not reliable since it is strictly dependent on the points used as CPs; if outliers or poor quality points are included in the CPs set, accuracy estimate is biased. In addition, when a low number of ground points is available, almost all of them are used as GCPs and very few CPs remain, so that RMSE may be computed on a poor (not significant) sample. In these cases accuracy assessment with the usual procedure is essentially lost. In addition, this method displays a low efficiency, making a poor use of the available information, as a large part of it must be collected and just used for validation purposes. The alternative method to the HOV to perform accuracy assessments of orthorectified HRSI is the Leave-One-Out cross-validation (LOOCV) method [13].

LOOCV was implemented into the software SISAR on the contrary in the world recognized commercial software OrthoEngine are required manual iterations for the LOOCV method therefore HOV was used in order to compare the results.

In Tables 4.2, 4.3 an example of results using LOOCV are presented; the tested images are acquired by QuickBird and Ikonos, they were oriented k times using all the known GPs as GCPs except one, different in each iteration, used as a CP, for every iteration the RMSE CP and median Absolute Deviation (mAD) were calculated; the mAD is a robust index and it is more correct to filter out the effect of the anomalous residual, not very representative of the mean achievable accuracy.

Table 4.2: Results in meter of LOOCV for Augusta *P001 - QuickBird

RMSE CPs							
label	**N[pix]**	**label**	**N[pix]**	**label**	**E[pix]**	**label**	**E[pix]**
1	-0.715	21	0.391	1	0.248	21	-0.787
2	-0.997	22	0.665	2	-1.398	22	-0.436
3	0.404	23	1.091	3	1.108	23	-0.282
4	-0.407	24	0.511	4	0.566	24	1.348
5	0.099	25	1.052	5	1.066	25	1.406
6	1.689	26	0.067	6	1.605	26	-2.011
7	1.811	27	0.611	7	-1.63	27	0.205
8	1.028	28	1.387	8	0.667	28	0.367
9	-0.864	29	0.017	9	-1.189	29	-0.743
10	0.263	30	-0.063	10	-0.468	30	0.019
11	0.519	31	0.186	11	1.168	31	2.563
12	0.605	32	-1.039	12	1.368	32	-0.338
13	0.108	33	-0.04	13	1.344	33	0.467
14	-1.262	34	-0.748	14	-0.293	34	-0.662
15	-0.309	35	-2.219	15	-0.599	35	-4.412
16	-0.106	36	1.214	16	-0.216	36	-1.759
17	-0.559	37	1.597	17	-1.337	37	3.896
18	-1.385	38	-1.98	18	1.251	38	0.287
19	-0.877	39	0.426	19	-1.749	39	1.005
20	-1.87			20	-1.012		
RMSE N[pix]				**RMSE E[pix]**			
0.998				1.438			
mAD N[pix]				**mAD E[pix]**			
0.665				1.012			

Table 4.3: Results in meter of LOOCV for Bagnoli3 - Ikonos

RMSE CPs							
label	**N[pix]**	**label**	**N[pix]**	**label**	**E[pix]**	**label**	**E[pix]**
1	-0.096	14	-0.168	1	0.331	14	-4.023
2	-0.438	15	-1.053	2	2.621	15	-1.962
3	-1.512	16	-0.133	3	-0.875	16	-0.718
4	2.357	17	0.545	4	-4.371	17	1.651
5	1.846	18	4.237	5	-2.207	18	-2.251
6	0.693	19	-0.311	6	-5.834	19	0.113
7	0.807	20	1.941	7	-2.578	20	-3.535
8	1.727	21	-1.656	8	-2.633	21	-2.781
9	-0.671	22	1.181	9	1.608	22	-1.354
10	0.904	23	-1.488	10	0.775	23	-0.966
11	-0.647	24	-3.054	11	0.454	24	0.884
12	3.273	25	1.404	12	2.739	25	-0.969
13	0.666			13	-5.933		
RMSE N[pix]				**RMSE E[pix]**			
1.665				2.672			
mAD N[pix]				**mAD E[pix]**			
1.053				1.962			

In this Section for each sensor the results of only one single image orientation are presented. The graphics of images are shown in the Appendix.

4.2 Rigorous model for single image (level 1A)

Twenty images acquired by different sensors were oriented with the two rigorous models implemented in SISAR and in OrthoEngine. In details

- 3 images of Rome acquired by EROS A (Figs.A.1, A.2, A.3)
- Salerno "joint", Rome images and Augusta stereopair acquired by QuickBird (Figs.A.4, A.5, A.6, A.7)
- Mausanne, Warsaw and Rome stereopairs acquired by Cartosat-1 (Figs.A.8, A.9, A.10, A.11, A.12, A.13, A.14, A.15)
- 4 stereopairs of Augusta (4505-4545) acquired by WorldView-1 (Figs.A.16, A.17, A.18, A.19, A.20, A.21)

Obviously in this Section the results are related to the orientation of single images only, therefore for the stereopairs single images are processed separately.

4.2.1 EROS A images

For Rome image (R2) the results in terms of RMSE on CPs are compared. RMSE trend are globally similar for both software and its value is comparable with the GSD, except for the North component (Fig.A.1). For the images of EROS A (R1, R4) the SISAR image orientation is satisfied for R4 image (around 2.0 m) in both components versus 4.0 m in East direction and 8.0 m in North direction in OrthoEngine (Fig.A.3); on the contrary SISAR performances is under expectation for R1 image, where RMSE CPs in OrthoEngine is close to GSD value versus 2.3-2.5 m for SISAR in East and North components respectively (Fig.A.2).

4.2.2 QuickBird images

For Salerno "joint" image the results in terms of RMSE on CPs are compared. RMSE trend are globally similar for both software and its value is comparable with the GSD (Fig.A.4).
In the image of Rome acquired by QuickBird the accuracy is similar for the two software but it is necessary to underline that this good results in SISAR software are obtained forcing the estimation of d_L parameters, discarded by SVD selection (Fig. A.5).

Regards Augusta images (*P001 and *P002) the CPs residuals in North component in SISAR software are less than the corresponding in OrthoEngine; the reverse behaviour is shown for the East component, where better results are devolved upon OrthoEngine software (Figs.A.6, A.7).

4.2.3 Cartosat-1 images

For Cartosat-1 Mausanne image the accuracy, both in bandA and in bandF, are close to 3.5-4.0 m, about 1 m upon the GSD of the images. For the bandA and the bandF the RMSE CPs is comparable between the two software in East component, instead in North component only in the Aft image SISAR software gives same results than OrthoEngine since in Forward image SISAR accuracy is lower (about 1.5 m) than OrthoEngine (Figs.A.8, A.9).
For this sensors the results of image orientation are changeable, in details for CastelGandolfo the accuracy is around 2.0-2.5 m in both components while in OrthoEngine the accuracy is around 1.5-2.0 m (Figs.A.14, A.15).
Better accuracy for SISAR software in East component for the image of Rome (bandA and bandF) (Figs.A.10, A.11) and for the image of Warsaw only Aft image respect to OrthoEngine (Fig.A.12), the behaviour reverses in North component where OrthoEngine have better accuracy for the Rome images (bandA and bandF) (Figs.A.10, A.11) except for Warsaw Forward image where OrthoEngine accuracy is decresed about 0.5-1.0 m respect to SISAR results (Fig.A.13).

4.2.4 WorldView-1 images

The WorldView-1 images are 4 scenes, two by two overlapped; the scenes have been stitched in order to have only two images that form one stereopair. In Augusta 4505 R1C1-R2C1 the orientation produces optimum results in SISAR software, around 0.5 m in East component and around 0.7 m in North component; little worse results in OrthoEngine (around 0.8 m) for both components (Fig. A.16). Analising the WorldView-1 4545 R1C1-R2C1, SISAR RMSE CPs has the same trend respect to 4505 image whereas OrthoEngine RMSE CPs is very high mainly in North component; this behaviour may be is due to the wrong modelling of OrthoEngine software of image acquisition mode; infact it is underlined that 4505 image is acquired in forward mode (from North to South) while 4545 image is acquired in reverse mode (from South to North) (Fig.A.17). The same behaviours are shown when the 4 WorldView-1 scenes are oriented separately without to be stitch. In details, worse accuracies are displayed out for the images acquired in reverse mode (4505 R1C1- 4505 R2C1) (Figs.A.18, A.19), whereas similar accuracies are shown in scenes 4545 R1C1 and 4545 R2C1 (Figs.A.20, A.21).

In the Tables 4.4 4.5, the results of the Level 1A single image orientation in meter and in pixel are summarized, in detail the RMSE CPs relative to the

number of GCPs, in which the accuracy is assesed, is carried for SISAR and OrthoEngine software.

Table 4.4: Results in meter of Level 1A single image orientation

RMSE CPs							
Sensor	**Area**	**SISAR**			**OrthoEngine**		
		n°GCPs	**E[m]**	**N[m]**	**n°CPs**	**E[m]**	**N[m]**
Level 1A images							
EROS A	Rome (R1)	17	2.39	2.51	17	2.06	1.91
	Rome (R2)	13	4.06	6.58	17	4.75	7.54
	Rome (R4)	13	2.39	2.43	17	2.52	3.06
QuickBird	Rome	17	0.44	0.47	13	0.54	0.51
	Augusta (*P001)	13	1.30	1.12	21	0.85	1.37
	Augusta (*P002)	13	0.96	0.94	13	0.93	1.06
	Salerno "joint"	17	0.52	0.84	21	0.72	0.65
Cartosat-1	Rome (bandA)	12	1.98	1.85	12	2.48	1.53
	Rome (bandF)	18	1.99	2.34	15	2.35	1.96
	CastelGandolfo (bandA)	21	2.44	2.59	21	1.96	2.07
	CastelGandolfo (bandF)	9	2.19	2.32	21	2.83	2.30
	Warsaw (bandA)	13	1.78	1.55	13	1.65	1.41
	Warsaw (bandF)	17	3.27	2.28	17	1.82	1.38
	Mausanne (bandA)	13	3.40	2.75	9	3.65	3.12
	Mausanne (bandF)	9	3.79	2.62	9	4.20	3.22
WorldView-1	Augusta (4505 R1C1-R2C1)	9	0.46	0.69	13	0.77	0.76
	Augusta (4545 R1C1-R2C1)	13	0.44	0.70	13	1.01	2.26

Table 4.5: Results in pixel of Level 1A single image orientation

RMSE CPs							
Sensor	**Area**	**SISAR**			**OrthoEngine**		
		n°GCPs	**E[pix]**	**N[pix]**	**n°CPs**	**E[pix]**	**N[pix]**
Level 1A images							
EROS A	Rome (R1)	17	1.33	1.39	17	1.14	1.06
	Rome (R2)	13	1.56	2.53	17	1.83	2.90
	Rome (R4)	13	1.26	1.28	17	1.33	1.61
QuickBird	Rome	17	0.72	0.77	13	0.89	0.84
	Augusta (*P001)	13	1.69	1.45	21	1.10	1.78
	Augusta (*P002)	13	1.28	1.25	13	1.24	1.41
	Salerno "joint"	17	0.78	1.25	21	1.07	0.97
Cartosat-1	Rome (bandA)	12	0.79	0.74	12	0.99	0.61
	Rome (bandF)	18	0.80	0.94	15	0.94	0.78
	CastelGandolfo (bandA)	21	0.98	1.04	21	0.78	0.83
	CastelGandolfo (bandF)	9	0.88	0.93	21	1.13	0.92
	Warsaw (bandA)	13	0.71	0.62	13	0.66	0.56
	Warsaw (bandF)	17	1.31	0.91	17	0.73	0.55
	Mausanne (bandA)	13	1.36	1.10	9	1.46	1.25
	Mausanne (bandF)	9	1.52	1.05	9	1.68	1.29
WorldView-1	Augusta (4505 R1C1-R2C1)	9	0.46	0.69	13	0.77	0.76
	Augusta (4545 R1C1-R2C1)	13	0.44	0.70	13	1.01	2.26

4.3 Rigorous model for single image (level 1B)

As example for the level 1B imagery, four images was selected, acquired respectively by Ikonos (Fig.A.22), QuickBird (Fig.A.23), GeoEye-1 (Fig.A.24) and WorldView-1 (Fig.A.25) satellites. All images represent the area of Rome. The images have been oriented with rigorous models implemented in SISAR and in OrthoEngine. Also in this case RMSE trend is similar for both software and accuracy is around the GSD value.

In the Tables 4.6, 4.7 the results of the Level 1B single image orientation in meter and in pixel are summarized, in detail the RMSE CPs relative to the number of GCPs, in which the accuracy is assesed, is carried for SISAR and OrthoEngine software.

Table 4.6: Results in meter of Level 1B single image orientation

RMSE CPs								
Sensor	**Area**	**Acquisition mode**	**SISAR**			**OrthoEngine**		
			n°GCPs	**E[m]**	**N[m]**	**n°GCPs**	**E[m]**	**N[m]**
Level 1B images								
Ikonos	Rome	Forward	10	0.86	0.72	10	0.97	0.74
GeoEye-1	Rome	Reverse	12	0.33	0.63	8	0.30	0.62
WorldView-1	Rome	Forward	8	0.26	0.51	10	0.57	0.37
QuickBird	Rome	Forward	10	0.54	0.30	10	0.49	0.38

Table 4.7: Results in pixel of Level 1B single image orientation

RMSE CPs								
Sensor	**Area**	**Acquisition mode**	**SISAR**			**OrthoEngine**		
			n°GCPs	**E[pix]**	**N[pix]**	**n°GCPs**	**E[pix]**	**N[pix]**
Level 1B images								
Ikonos	Rome	Forward	10	0.86	0.72	10	0.97	0.74
GeoEye-1	Rome	Reverse	12	0.66	1.26	8	0.60	1.24
WorldView-1	Rome	Forward	8	0.52	1.0	10	1.14	0.74
QuickBird	Rome	Forward	10	1.08	0.60	10	0.98	0.76

4.4 Rigorous model for stereopair (level 1A - level 1B)

Four examples of level 1A stereopair orientations and one examples of level 1B stereopair orientation using rigorous models are presented. The couple of EROS A images is an across-track stereopair; the EROS A stereopair is formed by across-track images, which have been acquired with a time delay of 1 year, and they have also a different spatial resolution (respectively 1.80 m and 2.60 m). The other four examples concern along-track stereopairs acquired by QuickBird, Cartosat-1, WorldView-1 and Ikonos satellites. Notice that Cartosat-1 is a satellite dedicated expressly to stereo viewing, having two CCD-line sensor cameras, looking respectively in forward direction with a nadir angle of 26° and in aft direction with a nadir angle of 5°. In the Ikonos stereopair the acqusition mode is reverse.

4.4.1 EROS A

SISAR results are compared again with OrthoEngine ones (Fig.A.26). The accuracies for the EROS A images are respectively at the level of 2.6 m (North) and 2.5 m (East) for SISAR and at the level of 4.6 m (North) and 2.6 m (East) for OrthoEngine. For the Height component the trend of the RMSE on CPs is similar for the two software, SISAR accuracy is around 5 m and OrthoEngine one is around 6.5 m.

4.4.2 QuickBird

For the QuickBird images the RMSE CP trend in North component is similar, although SISAR has the best accuracy; on the contrary in the East component the CP trend is different for the two software. In Height component the CP trend has a different value range for two software; for SISAR the accuracy varies between 0.8 and 1.1 m, instead for OrthoEngine it varies between 1.2 and 1.3 m (Fig.A.27).

4.4.3 Cartosat-1

For the Cartosat-1 stereopair of Rome RMSE CP trend in all components is similar, except for the East component where SISAR results are less than 2.0 m, while OrthoEngine ones are slightly worse (Fig.A.28). The stereopair orientation reflects the single image orientation infact good accuracy we have for Mausanne stereopair (around 1.0 m in planimetry and around 2.5 m in altimetry) in SISAR software, in OrthoEngine the trends are shifted respect to SISAR (Fig.A.29).

Both for Cartosat-1 image of Warsaw and CastelGandolfo OrthoEngine shows better results respect to SISAR (Figs.A.30, A.31).

4.4.4 WorldView-1

For WorldView-1 stereopair the RMSE CPs in all components is about constant and around the GSD value in planimetry and around 1.0 m in altimetry for SISAR software, while in horizontal component the accuracy range varias between 1.0 and 1.8 m and in vertical component the CPs residuals are around 4.0 m due to may be the image acquisition mode like described previously (Fig.A.32). Also in this case the orientation of the two tiled stereopairs of Augusta reflects the single image orientation, that means good accuracy for both stereopairs for SISAR software whereas for OrthoEngine the RMSE CPs in height is worse with respect to SISAR due to may be the different acquisition mode for the two images that composed the stereopair (Figs.A.33, A.34).

4.4.5 Ikonos

For the Ikonos stereopair for all components the RMSE CPs range is between 1.0 m and 2.0 m; the OrthoEngine software has a good accuracy in East and in height components respect to SISAR software, which has a lower RMSE CPs in North component (Fig.A.35).

In the Table 4.8 the results of the Level 1A and Level 1B stereopair orientation in meter are summarized, in detail the RMSE CPs (in East E, in North N and in Height H) relative to the number of GCPs , in which the accuracy is assesed, is carried for SISAR and OrthoEngine software.

Table 4.8: Results in meter of level 1A - level 1B stereopair orientation

RMSE CPs									
Sensor	**Area**	**GSD[m]**	**n°GCPs**	**SISAR**			**OrthoEngine**		
				E[m]	**N[m]**	**H[m]**	**E[m]**	**N[m]**	**H[m]**
Level 1A images									
EROS A	Rome (R1/R2)	3.10	21	2.50	2.81	5.94	3.17	4.40	6.97
QuickBird	Augusta (*P001/*P002)	0.77	21	0.50	0.84	0.86	0.78	1.07	1.33
Cartosat-1	Rome (bandA/bandF)	2.50	15	1.58	1.60	2.27	2.07	1.63	2.13
	CastelGandolfo (bandA/bandF)	2.50	21	1.79	1.57	5.12	2.26	2.04	1.32
	Warsaw (bandA/bandF)	2.50	13	1.83	1.59	2.18	1.64	1.09	2.01
	Mausanne (bandA/bandF)	2.50	9	2.03	2.65	2.95	2.44	3.85	4.02
WorldView-1	Augusta (4505/4545)	0.50	17	0.42	0.59	0.98	0.90	1.50	4.08
	Augusta (4045 R1C1- 4545 R1C1)	0.50	12	0.35	0.32	0.72	0.53	0.41	1.14
	Augusta (4045 R2C1- 4545 R2C1)	0.50	12	0.48	0.25	1.37	1.03	1.71	4.99
Level 1B images									
Ikonos	Bagnoli (Bagnoli3/Bagnoli4)	1.00	13	1.76	1.38	2.02	0.90	1.55	1.95

4.4.6 Examples of DSM extracted by SISAR software

One of the main application for satellite stereopair is the Digital Surface Models (DSMs) generation. DSMs have large relevance in many envariomental and engeneering applications, such as topographic mapping, spatial and temporal change detection, feature extraction, data visualization, etc. Elevation data may be acquired by several techniques, for example by LIDAR, optical and radar satellites as Shuttle Radar Topography Mission (SRTM) or ASTER and by aerial photogrammetry but DSMs extraction from satellite stereo pair offers some advantages, among which low cost, speed of data acquisition and processing, availability of several commercial software and algorithms for data processing. Generally in-track stereo imagery have remarkable advantages with respect to across-track stereo imagery, since the first acquisition mode reduces radiometric image variations, (temporal changes, sun illumination, etc.), and thus increases the correlation success rate in the image matching process, which is a crucial step for the automatic DSM production.

Hereafter we breafly summarize the path needed to generate DSMs without having to spell thinghs out:

- stereopair orientation, with rigorous model or RPCs model on the base of a number of GCPs
- image matching in order to point out the homologous points on the two images
- estimation of ground coordinates of all pixel derived from image matching
- interpolation of the data on a regular grid

At present, as already discussed, SISAR is able to manage orientation with rigorous or RPC model and to generate RPCs starting from its own rigorous model. No modules are implemented to perform image matching (they are presently under development).

Moreover, SISAR can produce the points cloud (ground coordinates) starting from image coordinates of the matched points, therefore, other software were used too.

In details the image matching was performed with DPCOR, whereas points cloud from the matched points was generated by RPCDEM (see Sec.5.5). Both these software were developed by Prof. K. Jacobsen, Leibniz University of Hannover.

DPCOR is imbedded in the measurement program DPLX allowing a fast check of the matched points. It is designed for the image matching, is using a least squares matching, respecting the tilt of the matching windows in object space, opposite to image correlation. The least squares image matching includes

an affine transformation of the sub-matrix of one image to the sub-matrix of the other image. In addition a constant shift and linear changes of the grey values with both coordinates are included, leading to 9 unknowns. The precise least squares matching has the disadvantage of a low convergence radius - the corresponding image positions must be known on a higher level. In DPCOR this is solved by region growing. Starting from at least one corresponding point, the neighboured points are matched. Such a seed point may be a control point, which has to be measured in any case manually. By matching the neighboured points the geometric relations are improved before going to the next neighboured points. DPCOR is always following the path with the highest correlation coefficient up to the complete coverage of the stereopair with corresponding points having a correlation coefficient above a chosen threshold.

The interpolations of data are executed with the scientific software LISA-BASIC, developed by Hannover University too; the interpolation method used is "Moving surface" that means: "Universal method with mostly smooth contour course in flat terrains. Mostly a high accuracy, lower tendency towards a formation of plateaus but also a bit slower. Automatic change between polynomials of 2nd order, tilted plane and horizontal plane. For special occasions, for example, if the input data are digitised contours, instead of the automatic switching the method can be fixed to tilted plane".

The accuracy of the DSM extracted, in terms of RMSE Z (or RMSE computed on the discrepances beetween the DSM extracted and the reference DSM) is analised using DEMANAL, again developed at Leibniz University Hannover. In this software the analysis for a dependency against the terrain inclination (α) and the height level are included.

Finally, also the DSM filtering to produce DEM was performed. It is well know that the Digital Surface Model (DSM) is the representation of the first reflective or visible surface, including trees, buildings, and any other features which is elevated above the ground. If between the height values on top of vegetation, especially trees and also buildings, points of the bare ground are available, such DSMs can be filterd to DEMs. This filtering has been made with the Hannover program RASCOR.

Two DSMs extracted are presented and analysed: the first is obtained by the Cartosat-1 stereopairs and covers the CastelGandolfo Area and the latter by the WorldView-1 stereopair covering the Augusta Area (Sicily). The characteristics of the stereopairs are summerized in the Tab.4.1

4.4.7 CastelGandolfo DSM

For the Castelgandolfo stereopair DSM has been generated using RPCDEM. The scenes are orientated with the RPCs model using SISAR RPCs. The DSM

extracted is compared with a precise reference DSM extracted by aerial photos with Erdas 9.1 software. The reference aerial DSM covers an area of about $85km^2$ in the centre of the scene, including forest parts and both open and urban areas (Fig.4.8). The image matching was executed with DPCOR.

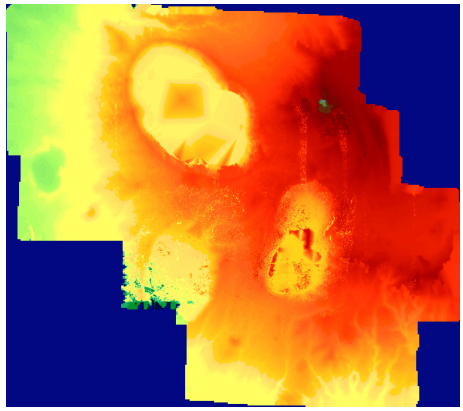


Figure 4.8: Height model extracted from aerial images

The matching was really good and the not matched points are mainly due to the lakes and to the clouds. The highest number of correlation coefficients is in the group r=0.90 up to 0.95; this distribution is due to the soil coverage (forest and lakes) and to the clouds on the image. The two software packages (Erdas 9.1 and RPCDEM) respond in different ways in the lake zone, in fact ERDAS applies an automatic filtering function so that it is able to assign an elevation even in the lake areas, on the contrary the matching achieved by DPLX did not recognize homologous points for the two images, due to the fact that no contrast was available on the lakes; therefore the elevation of those points was not extracted by RPCDEM.

Furthermore the accuracy of Cartosat-1 DSM has been checked, with respect to the reference DSM over different terrain types: open areas and urban areas; thus the overall scene has been divided in several selected regions. The analyses have been performed both for the DSM directly obtained from the images, and for the DEM, obtained filtering the original DSMs with the RASCOR.

Only the results of one open area and one urban area are presented. The selected open regions are not completely flat zones, because they still contain sparse buildings and groups of trees. Regarding the urban areas the discrepancies with the reference DSM are bigger; in these areas the smoothing effect of Cartosat-1 is more evident, in fact, while the DSM results higher than the reference over

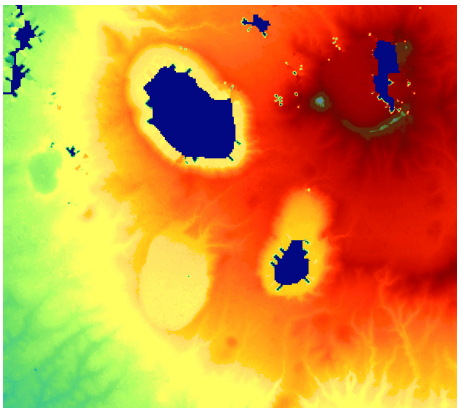


Figure 4.9: Height model extracted from Cartosat-1 images

the streets, it tends to smooth the edges of the buildings, as it is is shown by the profiles in Fig.4.10; on the contrary after filtering the profiles are similar and the smoothing effects disappear (Fig.4.11).

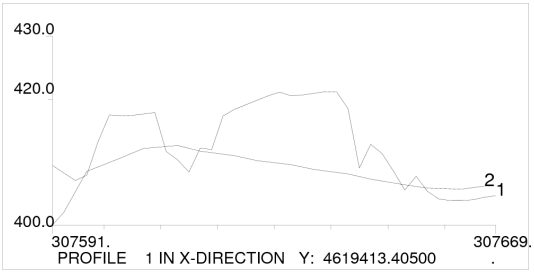


Figure 4.10: Profiles through a Cartosat-1 DSM (2) and the reference DSM from aerial images (1) in an urban area

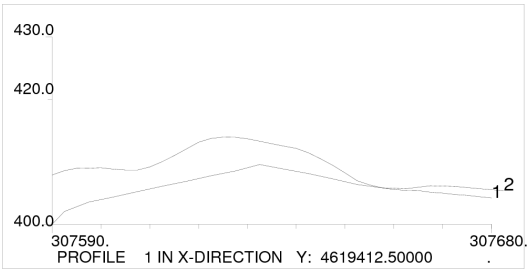
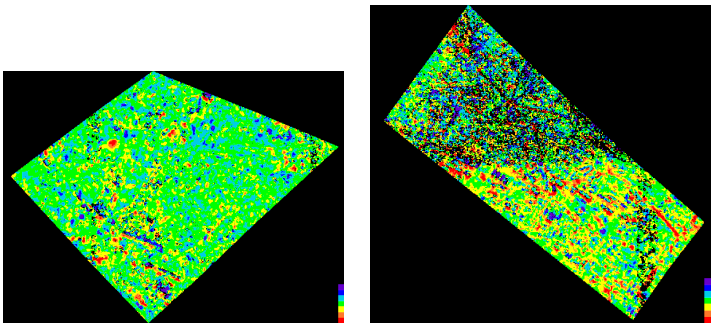


Figure 4.11: Profiles through a Cartosat-1 DEM (2) and the reference DEM from aerial images (1) in an open area



(a) (b)

Figure 4.12: Differential DSM of open area (a), and urban area (b)

Table 4.9: Colour coding referred to Fig.4.12

-10.5	-7.5	RED
-7.5	-4.5	BROWN
-4.5	-1.5	YELLOW
-1.5	1.5	GREEN
1.5	4.5	GREEN-BLUE
4.5	7.5	BLUE
7.5	10.5	VIOLETT

In the Fig.4.12 the differential DSMs, obtained by the comparison beetwen the DSM reference and the Cartosat-1 DSM generated with RPCDEM using the SISAR RPC, for chosen open area and urban area are shown; in Tab.4.10 the accuracy in terms of RMSE Z in the two selected areas (urban and open), both not filtering and filtering are presented. The accuracy of the DEM is better of the DSM one, the improvement is around 0.6 m for the two areas. In the open area the RMSE Z is the order of 2.5 m while it is worse in the urban area due to the matching difficultes suffered by the DPCOR matching algorithm.

Table 4.10: Accuracy of Cartosat-1 height models checked by precise reference DSM/DEM (*open area, **urban area)

Image	Area	SZ[m]	bias[m]	SZ =f(α=inclination)[m]
CastelGandolfo	no filtered	2.88*	-0.06	2.71+0.41*tan α
	filtered	2.29*	0.30	2.26+0.17*tan α
	no filtered	4.67**	-0.58	3.95+1.64*tan α
	filtered	4.06**	-0.34	3.27+1.91*tan α

4.4.8 Augusta DSM

As regard the DSM extracted from the Augusta stereopairs, the both stereopairs have been acquired in in-track mode, in addition, the two tiles, which form one scene of image, have been stitched and the two stitched images were oriented with the rigorous models implemented in SISAR software. The results of the orientation are summarized in Table 4.12.

The orientation test of the stiched WorldView-1 stereopair was performed also with commercial software OrthoEngine - PCI Geomatics v.10.2 but the precision of the model, checked by the RMSE of 13 GCPs, and the accuracy achievable evaluated in terms of RMSE on 19 CPs are not satisfing, so that the two stitched images were finally oriented separately. SISAR rigorous model produces better results than OrthoEngine, and the level of achievable accuracy is the same for the two images, while OrthoEngine results get worse especially in the WorldView-1 (4545 R1C1-R2C1) image in North component. A possible reason of this strange behaviour may be a not correct modelling of acquisition mode (as previously described).

In particular, the WorldView-1 4505 R1C1-R2C1 and 4545 R1C1-R2C1 images were acquired respectively in forward mode (from North to South) and reverse mode (from South to North). Thus a wrong modelling of reverse scan mode

Table 4.11: Results of single image orientations of WorldView-1 stereopair

13 GCP /19 CP	RMSE 13 GCPs			
	WorldView-1 4505		WorldView-1 4545	
	N[m]	E[m]	N[m]	E[m]
OrthoEngine	0.41	0.47	2.11	0.98
SISAR	0.36	0.38	0.61	0.37
13 GCP /19 CP	RMSE 19 CPs			
	WorldView-1 4505		WorldView-1 4545	
	N[m]	E[m]	N[m]	E[m]
OrthoEngine	0.76	0.77	2.26	1.01
SISAR	0.67	0.45	0.7	0.44

could cause the low precision and accuracy level in the image 4545 R1C1-R2C1 orientation.

In order to minimize the errors due to a not correct orientation model, the separated images were orientated with OrthoEngine. In this manner the model precision level is around 0.5 m.

The results of stereopair orientation in terms of RMSE on CPs are shown in Table 4.12; SISAR results are related to the stitched images, instead OrthoEngine to the north and the south tiles separately.

Table 4.12: Results of stereopair orientation

RMSE CP					
SISAR					
	n°GCP	n°CP	N[m]	E[m]	Height[m]
Stitched stereopair	13	19	0.58	0.39	0.98
OrthoEngine					
	n°GCP	n°CP	N[m]	E[m]	Height[m]
North stereopair	9	8	0.87	0.52	1.36
South stereopair	9	7	1.65	1.55	3.37

In this case image matching was performed with DPCOR but the points cloud was generated with SISAR software. Then, the extracted DSM has been managed in Hannover software LISA, able to visualize and interpolate the elevation data. In OrthoEngine two different DSMs, related to the North and South

areas, partially overlapped were generated; it is remarkable to underline that, in the overlapping area, the two DSMs have significant differences (the differences are close to 10 m). In order to give much continuity to the model in the shared area, four additional ground control points have been used for the DSM generation. The ground coordinates of additional points were computed in the SISAR software, using such points as TPs in the stereopair orientation, like this the differences between the two DSMs are reduced (around 1 m).

Two types of comparison were carried out where possible, in order to assess both average accuracy and its dependency on morphology and land cover types:

- sample comparisons, based on significant samples of independent check points with an average 3D accuracy of 0.3-0.5 m collected by kinematic GPS surveys
- local comparison with QuickBird DSM, extracted for a previously investigation with an accuracy of 2.5 m
- with WorldView-1 DSM extracted by OrthoEngine software

As regards the comparison between the DSM and the samples of CPs, it has to be underlined that the estimated accuracy does not consider some tens of points close to the DSM edges, where a considerable distortion (up to several tens of meters) is displayed, so some outliers have been eliminated through a standard rejection procedure of the normalized residual w_i [9].

$$|w_i| = \frac{|v_i|}{\sigma_v} < 3 \tag{4.1}$$

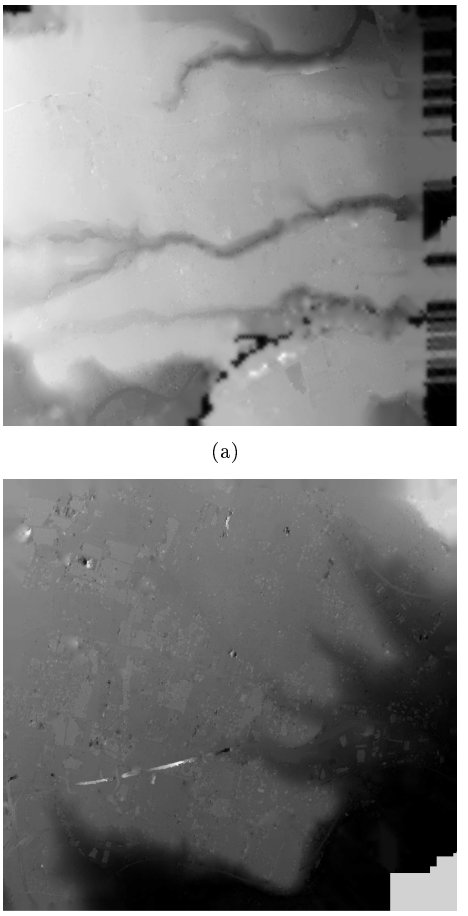
where σ_v is chosen equal to RMSE of height differences between DSM and the samples of CPs.

In order to evaluate the capability of DSM extracted from WorldView-1 satellite, with respect to DSM previously extracted from QuickBird on the same area (2.5 m average vertical accuracy), local comparison between two DSMs have been executed. To evaluate the accuracy of the DSM, four test areas have been selected; for the sake of brevity only the results related to one urban and one open area in the Northern part of the image are presented in Tab.4.13. Comparisons between three DSMs (DPCOR/SISAR WorldView-1, OrthoEngine WorldView-1, OrthoEngine QuickBird) and sample of CPs are listed according to their value of bias, standard deviation and RMSE Z.

DSM accuracy is at 2 pixel level, with respect to the CPs derived from GPS kinematic survey. These results may appear optimistic due to the special location of the CPs along roads. Overall, they are confirmed by the global comparison between SISAR WorldView-1 and OrthoEngine QuickBird DSMs.

Table 4.13: Results of accuracy assessment

	Open Area North		
	Bias [m]	St Dev [m]	RMSE Z [m]
DPCOR/SISAR WorldView-1-CPs	-0.60	1.14	1.29
OrthoEngine WorldView-1-CPs	1.32	0.57	1.44
OrthoEngine QuickBird-CPs	0.27	0.60	0.66
OrthoEngine QuickBird-DPCOR/SISAR WorldView-1	1.39	2.25	2.64
OrthoEngine QuickBird-OrthoEngine WorldView-1	0.06	2.11	2.11
DPCOR/SISAR WorldView-1-OrthoEngine WorldView-1	-1.69	1.90	2.52
	City North		
	Bias [m]	St Dev [m]	RMSE Z [m]
DPCOR/SISAR WorldView-1-CPs	0.21	0.59	0.63
OrthoEngine WorldView-1-CPs	0.34	0.64	0.72
OrthoEngine QuickBird-CPs	2.91	0.77	3.01
OrthoEngine QuickBird-DPCOR/SISAR WorldView-1	2.65	1.90	3.26
OrthoEngine QuickBird-OrthoEngine WorldView-1	2.26	1.83	2.91
DPCOR/SISAR WorldView-1-OrthoEngine WorldView-1	-0.29	1.56	1.59



(a)

(b)

Figure 4.13: DSMs extracted using DPCOR/SISAR on open area(a), urban area (b)

4.5 Rigorous model for stereopair (level 1B)

The results for Ikonos along-track stereopair are an example of the application of the stereo rigorous model to level 1B imagery. SISAR results are compared with OrthoEngine ones (Fig.A.35). In the North component SISAR results are better then OrthoEngine ones, whereas the opposite is true for the East component. In the Height component the accuracy trend is the quite similar for both software: SISAR accuracy varies between 1.4 m and 2.1 m, OrthoEngine accuracy varies between 1.2 m and 2.0 m.

Chapter 5

Application of the RPCs orientation models

SISAR results are analyzed and compared with the results obtained by commercial software:

- PCI OrthoEngine 10.0 (PCI Geomatics)
- Erdas 8.6 (Leica)

Results were presented at three different process levels: without adjustment, with a shift adjustment and with an affine adjustment. All three software (PCI only in the version 10.0) offers the possibility to refine RPF with a null order (shift) and a first order (affine transformation) correction. Results are obtained by increasing the number of GCPs from 1 to the maximum available for every image, for each test the GCPs set was selected so as they were well-distributed on the whole image. It has to be stressed that the most interesting are those obtained with few (e.g. 5) GCPs, since, otherwise, RPF model with RPCs becomes less and less competitive with respect to the rigorous one.

5.1 Results of RPCs application

Some companies (DigitalGlobe, SpaceImaging, etc.) managing satellites (QuickBird, Ikonos), supply with the images a set of related metadata files, among

which a file that contains the RPCs is included. In other cases (as EROS A, managed by ImageSat), no RPCs are supplied into the metadata, so that EROS A images are not considered in this Section. As regards RPCs application not all results are discussed, in detail:

- those one related to formal model precision (RMSE of residuals on GCPs - RMSE GCP) and no adjustment (that is, just the application of the metadata RPCs, without any additional correction) are showed for all images
- those one related to accuracy (RMSE of residuals on CPs - RMSE CP) are showed for the Ikonos Rome and Ikonos Bagnoli, QuickBird Rome, one Augusta (e.g. *P001) and one Salerno (e.g. *P002) images

For each image an histogram is depicted. It summarizes the RMSE CP, using 5 GCP, for three different software (SISAR, OrthoEngine, Erdas) when three different process levels (no adjust, shift, affine) are applied at (I, J) coordinates. Further each image was oriented with RPCs varying the number of available GCP and RMSE CP is presented. In this respect, note that the RMSE CP is really statistically significant if computed with an half of the available GPs at least, that is if GPs are split into two equivalent sets, one acting as GCPs and other acting as CPs. In the following graph the half of GPs is marked with a vertical bar. Please note that in the following graphs the ordinate scale is not the same in all images, in order to better focus each individual result.

At first it is important to remember that for all images oriented with 1 GCP by the shift transformation and with 3 GCPs by the affine one, the GCPs residuals and their RMSE must be zero. That behaviour is verified for SISAR and Erdas software but not for OrthoEngine (Tabs.5.1, 5.2, 5.3, 5.4, 5.5). This behavior do not have any mathematical interpretation. As regard the internal accuracy of the orientation, in case of the simple RPCs application without estimation of the refinement parameters all the available GPs may act as CPs. Without adjustment the RMSE of CP residuals are very high for QuickBird images ($\sim$ 5-55 pixel) with respect to Ikonos images ($\sim$2 pixel); as regards software performances, SISAR and OrthoEngine behave quite similar, at the level of few tenths to two pixels, whereas Erdas displays a very different behaviour. Also in this case, software should displays similar (if not identical) results, so that Erdas results have not any explanation (Tab.5.6). The estimation of refinement parameters and their application (shift and the affine transformations) remarkably improve the results described in Table 5.6.

Table 5.1: Ikonos Rome - Model precision for shift and affine transformation

Ikonos Rome - RMSE GCP - shift transformation						
n° GCP	SISAR		OrthoEngine		ERDAS	
	I [pix]	J [pix]	I [pix]	J [pix]	I [pix]	J [pix]
1	0.00	0.00	0.14	0.12	0.00	0.00
Ikonos Rome - RMSE GCP - affine transformation						
3	0.00	0.00	0.4	0.12	0.00	0.00

Table 5.2: Ikonos Bagnoli - Model precision for shift and affine transformation

Ikonos Bagnoli - RMSE GCP - shift transformation								
image	n° GCP	SISAR		OrthoEngine		ERDAS	
		I [pix]	J [pix]	I [pix]	J [pix]	I [pix]	J [pix]
Bagnoli1	1	0.00	0.00	0.07	0.10	0.00	0.00
Bagnoli2	1	0.00	0.00	0.08	0.07	0.00	0.00
Bagnoli3	1	0.00	0.00	0.04	0.09	0.00	0.00
Bagnoli4	1	0.00	0.00	0.33	0.20	0.00	0.00
Ikonos Bagnoli - RMSE GCP - affine transformation							
Bagnoli1	3	0.00	0.00	0.08	0.32	0.00	0.00
Bagnoli2	3	0.00	0.00	0.15	0.22	0.00	0.00
Bagnoli3	3	0.00	0.00	0.03	0.00	0.00	0.00
Bagnoli4	3	0.00	0.00	0.21	0.01	0.00	0.00

Table 5.3: QuickBird Rome - Model precision for shift and affine transformation

QuickBird Rome - RMSE GCP - shift transformation						
n° GCP	SISAR		OrthoEngine		ERDAS	
	I [pix]	J [pix]	I [pix]	J [pix]	I [pix]	J [pix]
1	0.00	0.00	6.64	2.52	0.00	0.00
QuickBird Rome - RMSE GCP - affine transformation						
3	0.00	0.00	4.84	1.83	0.00	0.00

Table 5.4: QuickBird Augusta - Model precision for shift and affine transformation

QuickBird Augusta - RMSE GCP - shift transformation							
image	n° GCP	SISAR		OrthoEngine		ERDAS	
		I [pix]	J [pix]	I [pix]	J [pix]	I [pix]	J [pix]
(*P001)	1	0.00	0.00	4.74	0.76	0.00	0.00
(*P002)	1	0.00	0.00	4.93	-0.84	0.00	0.00
QuickBird Augusta - RMSE GCP - affine transformation							
(*P001)	3	0.00	0.00	2.04	0.18	0.00	0.00
(*P002)	3	0.00	0.00	7.24	0.78	0.00	0.00

Table 5.5: QuickBird Salerno - Model precision for shift and affine transformation

QuickBird Salerno - RMSE GCP - shift transformation							
image	n° GCP	SISAR		OrthoEngine		ERDAS	
		I [pix]	J [pix]	I [pix]	J [pix]	I [pix]	J [pix]
(*P001)	1	0.00	0.00	-0.96	1.11	0.00	0.00
(*P002)	1	0.00	0.00	0.89	1.01	0.00	0.00
(*P003)	1	0.00	0.00	0.72	1.00	0.00	0.00
QuickBird Salerno - RMSE GCP - affine transformation							
(*P001)	3	0.00	0.00	0.63	1.11	0.00	0.00
(*P002)	3	0.00	0.00	0.53	0.60	0.00	0.00
(*P003)	3	0.00	0.00	0.56	0.19	0.00	0.00

Table 5.6: Results with no correction for all tested images

Sensor	Image	n° CPs	RMSE CP					
			SISAR		OrthoEngine		Erdas	
			I [pix]	J [pix]	I [pix]	J [pix]	I [pix]	J [pix]
Ikonos	Rome	27	2.23	2.24	2.88	2.72	14.26	3.06
	Bagnoli1	15	2.12	1.06	1.71	1.01	11.13	19.42
	Bagnoli2	15	1.84	1.01	2.11	1.00	13.84	19.27
	Bagnoli3	25	2.09	1.57	1.61	1.64	10.90	19.26
	Bagnoli4	25	2.09	1.26	2.34	1.33	13.93	19.25
QuickBird	Rome	49	49.34	17.79	49.91	17.32	49.92	14.38
	Augusta (*P001)	39	37.06	4.25	35.75	4.91	20.29	29.23
	Augusta (*P002)	38	53.33	9.27	54.04	7.37	57.84	20.32
	Salerno (*P001)	15	7.22	7.07	6.73	7.56	23.21	27.54
	Salerno (*P002)	34	7.31	9.29	6.67	9.60	23.90	30.22
	Salerno (*P003)	16	6.15	6.12	5.54	6.34	24.10	27.57

5.1.1 Ikonos

As regard the Rome Ikonos image, results using the shift transformation are equal between SISAR and OrthoEngine but they are still slightly bad for Erdas (Figs.B.1, B.2); this difference is removed in the affine transformation (Figs.B.1, B.3).

For all Bagnoli images the affine transformation removes the shifts of Erdas software respect to other software. There are still some differences with 3 GCPs, the minimum (no redundancy) for affine transformation estimation (Figs.B.4, B.5, B.6, B.7, B.8, B.9, B.10, B.11, B.12, B.13, B.14, B.15). For all the Ikonos images the shift correction to vendors RPCs appear the most effective, able to achieve an accuracy of 1-1.5 pixel with 2-3 GCPs.

5.1.2 QuickBird

As regard Rome image acquired by Quickbird, the accuracy for the simple RPCs application, without any correction, is very similar for the three software at the level of decades of pixels (Fig.B.16). On the contrary, if corrections are applied the results remarkably improve. For example, if the shift correction is applied, SISAR results are close to Erdas and OrthoEngine ones except in correspondence of few GCPs (until of 4 GCP) where OrthoEngine supplies worse results in I coordinate (Fig.B.17). With the affine transformation, for OrthoEngine the RMSE CP increases with respect to shift transformation, this is an unexpected behaviour; for OrthoEngine the I trend drifts away from the ones obtained with the remaining software (SISAR and Erdas) (Fig.B.18).

For the Augusta (*P002) image the results with shift transformation are equal for the three software with a difference of 7 pixel between the two coordinates (I, J) (Fig.B.20); with affine adjustment the I coordinates for OrthoEngine is "shifted" of about 7 pixel from the I coordinate extracted from Erdas and SISAR software, the difference is reduced to 0.5 pixel using from 13 to 45 GCP (Fig.B.21), what again does not make sense within a RPCs based orientation.

For the Salerno image (*P002), the results with shift transformation (in I coordinate) are slightly different for SISAR and OrthoEngine whilst for Erdas there is a noticeable difference (Fig.B.23). With the affine transformation the results are slightly better and closer for all software (Fig.B.24). For all the QuickBird images the affine correction to vendors RPCs appear the most effective, able to achieve an accuracy of 1- 1.5 pixels with 4-5 GCPs.

5.2 Conclusions for RPCs application

SISAR software for the RPCs usage provides quite good results for all the tested images, with outcomes at least equal to those coming from the analyzed commercial software, often better with respect to at least one software. Thus, SISAR algorithm for RPCs application comes out as a valid methodology to use RPCs supplied by the image vendor companies. The utilized adjustment methods allow a rapid and useful improvement of the results. The adjustment level (shift and affine) depends on the satellite. In general, just a shift adjustment eliminates the most of the bias and an affine transformation is advisable for QuickBird images; on the contrary for Ikonos images a simple shift translation is suited. This is probably due to the different processing level of the imagery. Achievable image accuracy does not increase considerably more than 2-5 GCP for the shift transformation and more than 4-5 for the affine transformation; this fact permits to obtain an orientation of good quality even with a few GCPs, which is the main goal when RPCs are used. The behaviours of the tested commercial software vary, depending on the different corrections utilized; just to summarize:

- OrthoEngine displays a strange behavior with non-zero residuals when correction parameters are estimated with no redundancy
- the simple RPCs application with Erdas produces results quite different generally worst from those obtained with SISAR and OrthoEngine
- OrthoEngine accuracy is similar to SISAR in the simple RPCs application and consequently with the shift transformation, while OrthoEngine results worsen considerably with the affine transformation; in most of the tests the difference is more evident in the sample I direction.

5.3 Results of RPCs generation

The RPCs generated by SISAR software are organized in an appropriate file; in particular, for QuickBird images the RPCs and the normalized parameters are written according to already existing format file (.RPB file) and for EROS A the same structure of RPB QuickBird file is created.

The RPCs algorithm generation implemented in SISAR software produces an output file that can be used by the commercial software where only the estimable and significant RPCs are presented and the remaining RPCs are fixed to zero. In this way it is possible to orientate an image with the RPCs although the vendors RPCs are not available; this is the case of EROS A image and of Salerno stitch image where the RPCs for the single images are available but they are not available for the stitched image.

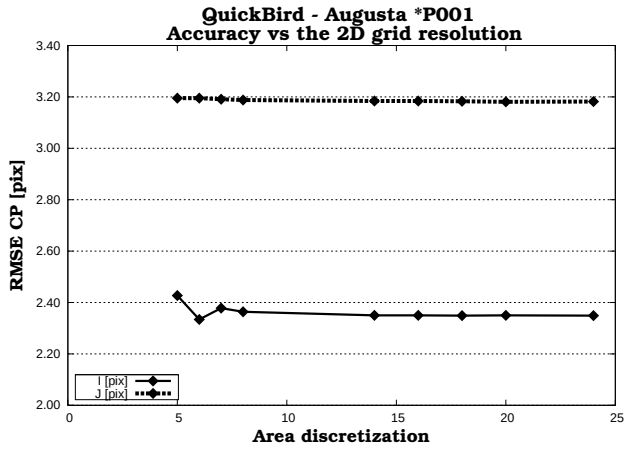


Figure 5.1: Accuracy trend increasing the 2D grid discretization (with 9 GCPs)

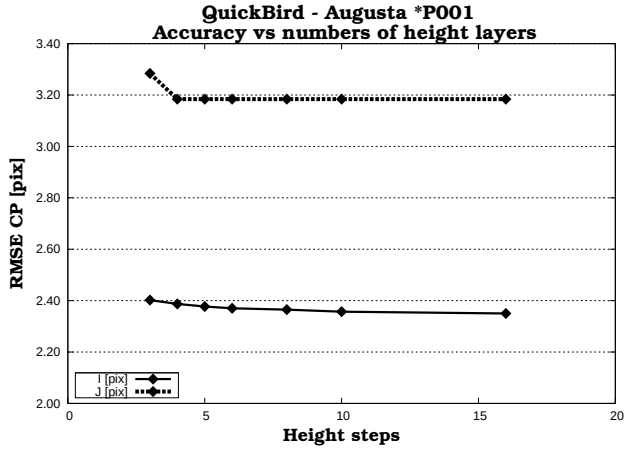


Figure 5.2: Accuracy trend increasing the number of layers (with 9 GCPs)

For the 3D grid generation and consequently for RPCs generation only results obtained with the minimum number of GCPs necessary to achieve a good accuracy stabilization using the rigorous sensor model [31] were considered. In the experiments a 15x15x15 discretization was used since the accuracy achieved does not improve for finer 2D grids; an example for QuickBird is outlined in Fig.5.1; actually the accuracy is quite stable starting from a 8x8 2D discretization.

The terrain morphology was quite plane in the images used for the tests, thus the elevation discretization was chosen the same, l=15; with l>15 the same achievable accuracy was preserved. The results for the same example for QuickBird are sketched in Fig.5.2. The vertical discretization could become important if the zone under concern is mountainous and the image elevation range is about thousand of meters.

The images chosen for the test are presented in the following Table 5.7, where it is reported the number of GCPs used to orientate the image with SISAR software and the number of RPCs really estimable and necessary for the orientation with RFM (SISAR RPCs). It may be suddenly noted that, as easily expected, that the number of RPCs is higher for the images with high off-nadir angle than the others (Tab.5.7).

Table 5.7: Number of SISAR RPCs vs Off-Nadir Angle

satellite	image	n°GCP	n° SISAR RPCs	Off-nadir (°)
QuickBird	Augusta(*P001)	13	41	29.2
	Augusta(*P002)	13	36	28.2
	Rome	17	26	3.0
	Salerno (*P001)	15	25	19.0
	Salerno (*P002)	17	23	20.0
	Salerno (*P003)	18	24	20.9
	Salerno "joint"	17	23	20.0
EROS A	R1	17	21	9.25
	R2	13	32	35.55
	R4	29	22	15.85
Ikonos	Rome	10	14	18.7
	Bagnoli1	8	28	22.98
	Bagnoli2	8	22	24.27
	Bagnoli3	12	25	22.98
	Bagnoli4	12	25	24.27

The SISAR RPCs were tested in SISAR, OrthoEngine and Erdas software,

except for those generated for EROS A which were not accepted by commercial software, probably because RPCs are not previewed for EROS A imagery. In detail, at first the comparison among the three software was done considering the results obtained without any correction; then corrected SISAR RPCs were used again in three software, SISAR, OrthoEngine and Erdas, and the results were compared with the ones obtained using the RPCs supplied by vendors (ISD RPCs). For all the images, the application of SISAR RPCs without correction shows a better accuracy with respect to the original RPCs, what may be easily explained since SISAR RPCs are generated starting from a rigorous model orientation based on high quality GCPs: the RMSE CP remarkably decrease with SISAR and OrthoEngine software, but not with Erdas software, where usually is increased (Tab.5.6 vs Tab.5.8). For the images without vendors RPCs (Salerno "joint", R1, R2, R4) it is not possible to do a comparison, so that only SISAR results are showed.

Table 5.8: Results with no adjustment for all tested images

Sensor	Image	n° CPs	RMSE CP					
			SISAR		OrthoEngine		Erdas	
			I[pix]	J[pix]	I[pix]	J[pix]	I[pix]	J[pix]
QuickBird	Rome	49	0.69	0.82	0.81	0.80	6.60	3.23
	Augusta (*P001)	39	2.01	3.00	2.93	3.69	18.02	28.18
	Augusta (*P002)	38	1.46	2.01	1.61	3.79	4.80	31.29
	Salerno (*P001)	15	1.59	2.09	1.59	2.05	16.17	20.28
	Salerno (*P002)	34	1.31	0.97	0.90	1.20	17.68	21.61
	Salerno (*P003)	16	1.46	0.92	1.01	1.15	19.09	22.61
	Salerno "joint"	57	1.41	1.72				
EROS A	R1	49	1.19	1.42				
	R2	49	1.92	2.62				
	R4	49	1.40	1.06				
Ikonos	Rome	27	0.72	0.62	0.93	0.72	0.81	0.65
	Bagnoli1	15	1.07	1.61	1.00	1.61	19.67	12.63
	Bagnoli2	15	0.95	1.33	0.88	1.41	19.64	15.28
	Bagnoli3	25	1.79	1.22	1.82	1.34	19.12	12.35
	Bagnoli4	25	1.56	1.43	1.53	1.75	19.42	15.78

5.3.1 Comparison between ISD RPCs and SISAR RPCs in SISAR software

For QuickBird Rome image, the ISD RPCs produce, in both components, worse results than SISAR RPCs using only shift transformation (Fig.B.25); instead, using an affine transformation the results improve in J direction (ISD RPCs) remaining around the same values in I direction (Fig.B.26). In Augusta (*P001) and Salerno (*P003) the affine transformation removes the behaviors differences (Figs.B.28, B.36) that are obtained using a shift transformation in both direction (Figs.B.27, B.35), this observation is valid also for Salerno (*P001), but just in I direction (Figs.B.31, B.32). Regards Augusta (*P002) image, in J direction the RMSE CP trend is similar using two types of RPCs in both adjustments; for the I direction with SISAR RPCs, the CP residuals are stable around 2m while the significant improvement is underlined using ISD RPCs in affine transformation (Figs.B.29, B.30). Both Salerno (*P002) image acquired by QuickBird and Rome image acquired by Ikonos the affine adjustment does not improve significantly the results (Figs.B.33, B.34, B.39, B.40).

Bagnoli1 and Bagnoli2 images using the two types of RPCs have the same trend but different RMSE CP: higher values for J component and lower ones for I components with SISAR RPCs (Figs.B.39, B.40, B.41, B.42). For Bagnoli3 and Bagnoli4 small differences are shown and same results are obtained using affine transformation (Figs.B.43, B.44, B.45, B.46).

5.3.2 Comparison between ISD RPCs and SISAR RPCs in OrthoEngine software

For the QuickBird Rome, Augusta (*P001), Augusta (*P002) images, both with shift and affine transformations, SISAR RPCs perform better than ISD RPCs; the accuracy range vary between 1 and 2 pixel when the accuracy is stabilized. It has to be accounted that the number of SISAR RPCs is significantly lower than ISD RPCs (about 1/2, 1/3 coefficients). SISAR performs better ISD mainly in I direction. Infact the results for the two kinds of RPCs are more similar for J component, but for I component the ISD RPCs with 1 GCP for the shift transformation point out high RMSE CP (Figs.B.47, B.51); the same situation happens also with of the minimum number of points (3 GCP) for the affine transformation (Figs.B.48, B.50, B.52). This behaviour is not noticed for the Augusta (*P001) in the shift transformation (Fig.B.49). In the Salerno (*P001) image the results with ISD RPCs are better than the SISAR RPCs ones. This image (*P001) is the upper in the Salerno strip, in this case the ground points distribution (Fig.4.3) is not optimal because the image is covered by forest for the most part, that has an impact to the results (Fig.B.53); they improve with

affine transformation (Fig.B.54). For Salerno (*P002) image the affine transformation does not improve the results, the RMSE CP remains nearly identical (Figs.B.55, B.56). On the contrary in the Salerno (*P003) image the results with ISD RPCs improve with the affine correction (Figs.B.57, B.58). For the Rome image, acquired by Ikonos, the results are similar in the two software using both transformations (Figs.B.59, B.60). For the Bagnoli images the RMSE CP, in OrthoEngine software, has the same trend that in SISAR software, therefore the discussion and considerations previously proposed in A are still valid (Figs.B.61, B.62, B.63, B.64, B.65, B.66, B.67, B.68).

5.3.3 Comparison between ISD RPCs and SISAR RPCs in Erdas software

In comparison with the Erdas software to OrthoEngine does not display a high CP residuals when the shift parameters are estimated for the Rome image (Fig.B.69). For both shift and affine transformation, in J direction the ISD RPCs and the SISAR RPCs show a similar trend, whilst in I direction there is a remarkable difference, being SISAR RPCs better (Figs.B.69, B.70). For the Augusta (*P001) image for both direction and with shift transformation, again SISAR RPCs are better then ISD RPCs; nevertheless these differences disappear with the first order affine refinement (Figs.B.71, B.72). An anomalous behavior has the Augusta (*P002) image in Erdas software compared to OrthoEngine software: with affine adjustment, I component of SISAR RPCs displays a worse accuracy with respect to ISD RPCs (for OrthoEngine software the behaviour is opposite). In J direction the equal results both the two software and both the ISD and SISAR RPCs are showed in shift adjustment, with small worsening in the affine transformation just when 3 GCPs are used (Figs.B.73, B.74). As regard Salerno (*P001) OrthoEngine and Erdas software have a similar behaviors (Figs.B.75, B.76). The affine transformation does not improve the SISAR RPCs results for the I coordinate for the Salerno (*P002) with remain slightly worse than ISD RPCs ones (Figs.B.77, B.78). In the Salerno (*P003) image the results are comparable only in I component for the shift adjustment, while in both coordinates (I, J) and in affine transformation the results are similar (Figs.B.79, B.80). For the Bagnoli images the RMSE CP, in Erdas software, has the same trend that in SISAR and OrthoEngine software, therefore the previously considerations are valid once again (Figs.B.83, B.84, B.85, B.86, B.87, B.88, B.89, B.90).

5.4 Summarizing results of comparison between SISAR RPCs and ISD RPCs

In order to summarize the results of the comparison between RPCs generated by SISAR software (SISAR RPCs) and the RPCs supplied by vendors (ISD RPCs) the ratio R of the RMSE CP was introduced as reference index:

$$R = \frac{\text{RMSECP(ISDRPCs)}}{\text{RMSECP(SISARRPCs)}} \tag{5.1}$$

This index was been computed, in all the testes, using 2 GCPs when the shift transformation is applied (Tab.5.9) and 5 GCPs for the affine transformation (Tab.5.10). This choice is justified considering that the strong advantage of using RPCs is the remarkable reduction of the needed GCPs.

In this respect it has to be noted that overall SISAR RPCs performs better than ISD RPCs with all the software and all the images, except for the affine correction with Ikonos. In the following tables the RMSE CP using the SISAR RPCs in all software with 2 GCPs in shift transformation and 5 GCPs in affine transformation are presented (Tabs.5.11, 5.12).

The results underline that with the SISAR RPCs, in all tested software, the average of accuracy, computed in terms of RMSE CP, is around 1.5 pixel both in QuickBird and in Ikonos images with the shift transformation, otherwise it is around 1 pixel in QuickBird images and more 1.5 pixel in Ikonos images with the affine transformation; this behavior remarks what is proposed previously (Tabs.5.9, 5.10).

Table 5.9: Ratio R in all tested software using shift transformation

RMSE CP(ISD RPCs)/RMSE CP(SISAR RPCs)							
Shift transformation	Image	OrthoEngine		Erdas 8.6		SISAR	
		I	J	I	J	I	J
QuickBird	Rome	0.19	1.04	0.5	0.86	0.64	0.79
	Augusta (*P001)	0.88	0.76	0.68	0.78	0.82	0.70
	Augusta (*P002)	0.27	1.09	0.38	1.36	0.33	1.18
	Salerno (*P001)	2.25	1.49	2.10	1.28	2.90	1.15
	Salerno (*P002)	1.01	0.41	1.28	0.54	1.32	0.51
	Salerno (*P003)	1.12	0.22	1.01	0.32	0.96	0.23
	Average	0.96	0.84	1.01	0.85	1.16	0.76
Ikonos	Rome	0.87	0.92	0.89	0.75	0.89	0.89
	Bagnoli1	0.88	1.10	0.87	1.13	0.91	1.14
	Bagnoli2	0.84	1.11	0.81	1.06	0.83	1.12
	Bagnoli3	1.03	0.87	0.99	1.08	1.02	0.89
	Bagnoli4	1.02	0.83	0.97	0.94	1.00	0.81
	Average	0.93	0.97	0.9	0.99	0.93	0.97

Table 5.10: Ratio R in all tested software using affine transformation

RMSE CP(ISD RPCs)/RMSE CP(SISAR RPCs)							
Affine transformation	Image	OrthoEngine		Erdas 8.6		SISAR	
		I	J	I	J	I	J
QuickBird	Rome	0.17	0.81	0.75	0.95	0.75	0.94
	Augusta (*P001)	0.93	1.16	1.09	1.16	0.96	1.14
	Augusta (*P002)	0.46	1.03	1.26	1.14	1.17	1.11
	Salerno (*P001)	1.28	1.91	0.82	1.81	0.80	1.73
	Salerno (*P002)	0.95	0.82	1.14	0.79	1.12	0.77
	Salerno (*P003)	0.75	0.61	1.13	0.93	1.14	0.93
	Average	0.82	1.07	1.27	1.12	1.11	1.10
Ikonos	Rome	1.02	1.07	1.01	1.07	1.01	1.10
	Bagnoli1	1.25	1.11	1.27	1.11	1.28	1.11
	Bagnoli2	1.13	1.14	1.15	1.14	1.15	1.14
	Bagnoli3	1.05	1.09	1.06	1.23	1.06	1.23
	Bagnoli4	1.08	1.06	1.10	1.13	1.10	1.12
	Average	1.11	1.10	1.12	1.14	1.12	1.14

Table 5.11: RMSE CP (SISAR RPCs) in all tested software using shift transformation

Shift Transformation		RMSE CP (SISAR RPCs)						
		GSD[m]	OrthoEngine		Erdas 8.6		SISAR	
			I[pix]	J[pix]	I[pix]	J[pix]	I[pix]	J[pix]
QuickBird	Rome	0.61	0.89	1.29	0.76	1.57	0.92	1.38
	Augusta (*P001)	0.77	1.82	1.44	1.83	1.45	1.79	1.30
	Augusta (*P002)	0.75	2.18	1.23	2.70	1.30	2.23	1.14
	Salerno (*P001)	0.67	1.88	2.05	1.99	2.24	1.98	2.08
	Salerno (*P002)	0.67	0.95	0.79	1.50	0.83	0.99	0.79
	Salerno (*P003)	0.67	0.96	0.36	1.28	0.47	0.99	0.37
	Average		1.45	1.21	1.78	1.37	1.49	1.22
Ikonos	Rome	1.00	0.75	0.66	0.81	0.70	0.74	0.67
	Bagnoli1	1.00	1.01	1.70	1.16	1.74	1.05	1.76
	Bagnoli2	1.00	0.93	1.47	1.04	1.39	0.94	1.47
	Bagnoli3	1.00	1.88	1.84	1.78	2.25	1.88	1.86
	Bagnoli4	1.00	1.90	2.59	1.88	3.07	1.92	2.65
	Average		1.29	1.65	1.33	1.83	1.31	1.68

Table 5.12: RMSE CP (SISAR RPCs) in all tested software using affine transformation

Affine Transformation		RMSE CP (SISAR RPCs)						
		GSD[m]	OrthoEngine		Erdas 8.6		SISAR	
			I[pix]	J[pix]	I[pix]	J[pix]	I[pix]	J[pix]
QuickBird	Rome	0.61	0.70	0.88	0.80	1.16	0.80	1.16
	Augusta (*P001)	0.77	2.15	1.32	2.15	1.29	2.13	1.27
	Augusta (*P002)	0.75	1.75	1.00	1.93	0.97	1.78	0.98
	Salerno (*P001)	0.67	0.66	1.73	0.56	1.96	0.56	1.93
	Salerno (*P002)	0.67	0.76	0.96	0.76	0.98	0.75	0.96
	Salerno (*P003)	0.67	0.94	0.53	1.18	0.54	1.19	0.54
	Average		1.16	1.01	1.23	1.05	1.20	1.04
Ikonos	Rome	1.00	0.90	0.78	0.91	0.82	0.91	0.83
	Bagnoli1	1.00	1.21	1.94	1.21	1.95	1.21	1.95
	Bagnoli2	1.00	1.06	1.58	1.06	1.59	1.06	1.58
	Bagnoli3	1.00	1.71	1.94	1.72	2.24	1.72	2.24
	Bagnoli4	1.00	2.19	2.12	2.24	2.42	2.24	2.41
	Average		1.41	1.67	1.43	1.80	1.43	1.80

5.4.1 Comparison between RPCs model and Rigorous model

For the EROS A images a comparison of SISAR software vs. OrthoEngine and Erdas software is not possible because in these commercial software a module for the orientation of EROS A images with RPF is not implemented; moreover, for the Salerno joint image a comparison between SISAR and Erdas is not possible because the used version of Erdas do not elaborate image huger than 2 GB.

Therefore, a comparison between the RPCs model, using SISAR RPCs with the shift transformation, and OrthoEngine rigorous model is presented. For two images (R1 and R2) the trend between two model is similar but the rigorous model has a slightly better accuracy than RPCs model with more than 15 GCPs, otherwise, RPCs model performs better (Figs.B.91, B.92).

For the R4 image the accuracy trend, in I and J components, obtained with the RPCs model is much more stable and it underlines accuracy range lower. It is possible explain this behavior if it considers that SISAR rigorous model, that is at the base of the generation RPCs, is more refined with respect to OrthoEngine rigorous model for the images with high off-nadir angle (Tab.4.1, Fig.B.93).

In case of the Salerno stitch the greater differences between the two model are showed in J direction, the RMSE CP trend with RPCs model is more consistent than RMSE CP trend obtained with rigorous model (Fig.B.94).

The RPCs model is comparable with respect to OrthoEngine rigorous model; the results, previously described, underline better accuracies when the number of available GCP is few (Fig.B.94), what is useful both for logistic and budget motivations.

5.4.2 Conclusions for SISAR RPCs generation

The SISAR RPCs can be used by commercial software (OrthoEngine and Erdas), the results, previously described, demonstrate that commercial software results using SISAR RPCs are absolutely comparable with the results obtained using ISD RPCs in the same software and globally even better, provided a proper correction (shift or affine) is used (Tabs.5.9, 5.10). It is underlined that SISAR RPCs are less than ISD RPCs (about $1/2$,$1/3$ coefficients). This comparison validates the quality of SISAR RPCs, the feasibility and stability of the SISAR RPCs generation process. The results demonstrate that SISAR RPCs permit high performance for image orientation, achieving a very good accuracy. The refinement models (shift and affine) do not always improve the results obtained by using SISAR RPCs for QuickBird images. Moreover it was demonstrated as the simple shift adjustment eliminates almost the totality of RPCs extraction errors. Hence, the affine model is rarely useful with SISAR RPCs. The RPCs results are directly dependent to the Rigorous model from which are extracted. Of note, EROS A and joint images do not provide RPCs with image files, so

that SISAR RPCs permit the only powerful orientation with few GCPs for this satellite.

5.5 Stereo model via RPCs

Finally an example of stereopair orientation via RPCs are presented. Results of SISAR software are compared with the Erdas ones and with the results obtained using a scientific software developed by Prof. K. Jacobsen of Leibniz University of Hannover. The module of the computation of ground coordinates of the corresponding image points, determined in original space images and based bias corrected RPCs is RPCDEM, included in the program system BLUH. The image coordinates of the corrisponding points on the two images, the orientation information of the scenes and some GCPs are necessary to carryout the processing. In detail the two scenes are oriented using RPCs by RAPORIO (BLUH module to the orientation of the original space images based on RPCs). The input file of RAPORIO are the ground control point coordinates (East, North and height above the WGS84 ellipsoid), the scene coordinates and the corrisponding file with rational polynomial coefficients. In the software it is possible select different options:

- the value of sigma a priori
- the number of the points can be excluded from the computation by the sigma a priori
- the bias correction of the data set (affine or shift correction)

In the test shown, the sigma a priori is chosen equal to 1, no point is eliminated and all the correction parameters (shift, scale and rotation in both direction) are estimated. The output file are: the improved object coordinates, then transformed in image coordinates using the nominal GSD, and the file for the data transfer to RPCDEM. Summarizing in the RPCDEM the ground points are computed by the intersection of the equations after the orientation separately of both scenes by the program RAPORIO.

The stereopair used is acquired by Cartosat-1 satellite on the area of Mausanne. It is oriented using SISAR, Erdas v.9.0 and RPCDEM software. The results obtained show equal RMSE CPs trend for all tested software both in North and East direction and in Height (RMSE CPs is around GSD value in East and North and around 4m in Height) (Figs.B.101, B.102) The behavior is the same for Cartosat-1 CastelGandolfo image for the three software (the range of RMSE CPs varies between 1.2 m and 1.8m in North component and 2.8 m and 1.8 m in East component and near to 1.4 m in height) (Figs.B.95, B.96) while in Cartosat-1 Rome image the trend in only similar with little shift among the three software (Figs.B.97, B.98); for the Cartosat-1 Warsaw image, in North direction SISAR shows a worse accuracy in North, it is shifted of 0.2 m respect to tha accuracy trend obtained with Erdas and RPCDEM and also in the Height

direction the RMSE CPs for SISAR is more high respect to Erdas and RPCDEM (Figs.B.99, B.100).

Table 5.13: Results in meter of stereopair orientation using RPC model

Sensor	Area	RMSE CPs			
		SISAR			
		n° GCPs	E[m]	N[m]	Height[m]
Cartosat-1	Rome	18	1.90	1.73	2.22
	Mausanne	17	3.03	2.59	3.77
	CatelGandolfo	12	1.53	1.19	1.29
	Warsaw	13	1.60	1.06	1.87
		Erdas			
		n° GCPs	E[m]	N[m]	Height[m]
Cartosat-1	Rome	18	1.89	1.62	1.83
	Mausanne	17	2.69	2.71	3.89
	CatelGandolfo	12	1.21	1.53	1.29
	Warsaw	13	1.29	1.16	2.34
		RPCDEM			
		n° GCPs	E[m]	N[m]	Height[m]
Cartosat-1	Rome	18	1.94	1.59	2.04
	Mausanne	17	2.72	2.69	3.91
	CatelGandolfo	12	1.51	1.22	1.28
	Warsaw	13	1.29	1.11	1.42

Chapter 6

Conclusions

High resolution satellite imagery (HRSI) became available in 1999 with the launch of IKONOS, the first civil satellite offering a spatial resolution of 1 m. Since then other high resolution satellites have been launched, among which there are EROS A (1.8 m), QuickBird (0.61 m), Orbview-3 (1 m), EROS-B (0.7 m), Worldview-1 (0.5 m) and GeoEye-1 (0.41 m) and others are planned to be launched in the next few years. Moreover many new satellites dedicated to stereo viewing, for example Cartosat-1 (2.5 m), have been launched.

HRSI have relevant impact in cartographic applications, as possible alternative to aerial imagery to produce orthophotos, to generate of Digital Elevation Models (DEM) and Digital Surface Models (DSM) and also for 3D feature/object extraction (e.g. for city modeling), especially in areas where the organization of photogrammetric flights may be critical.

The geomatic utilizations of satellite imagery require a high level geometric correction through image orientation. Two different types of orientation models are usually adopted: the physical sensor models (also called rigorous models) and the generalized sensor models.

In this thesis many features of the sensor models were discussed, both for single images and stereopairs. In details, discussions were focused on the rigorous models for the orientation of the basic image (level 1A) and of the image projected onto a specific object surface (usually an expanded ellipsoid; derived from the WGS84) (level 1B), as well on the RPC model; both models were extended to stereopairs.

The models, designed for the orientation of imagery acquired by pushbroom

sensors carried on satellite platforms, EROS A, QuickBird, Ikonos, Cartosat-1, WorldView-1 and GeoEye-1, were implemented in the software SISAR (Software per Immagini Satellitari ad Alta Risoluzione) developed at the Area di Geodesia e Geomatica - Sapienza Università di Roma.

After the presentation of the models, many applications were presented; to point out the effectiveness of the new models, SISAR results were compared with the corresponding ones obtained by different commercial software (OrthoEngine, Erdas) and by an other scientific software developed at Leibniz University of Hannover (Germany).

Real applications are demonstrated that rigorous models both for level 1A and level 1B imagery can provide orientation accuracy at 1-1.5 pixel level in the horizontal components (Tabs.4.5, 4.7), and at 1-2 pixel level in the height for stereopairs (even better with Cartosat-1 and slightly worse with EROS-1) (Tab.4.8).

The SISAR software for the RPCs usage provided optimal results in all tested imagery, with outcomes at least equal to the commercial software analyzed. Achievable image accuracy does not increase considerably even if more than 2-5 GCPs for the shift transformation and more than 4-5 for the affine transformation are used; this fact permits to obtain an orientation of good quality even with a few GCPs, which is the main goal when RPCs are used.

The RPCs generated by SISAR can be used by commercial software (OrthoEngine and Erdas), the results demonstrated that commercial software results using are comparable with the ones obtained using metadata (ISD) RPCs in the same software and globally even better, provided a proper correction (shift or affine) is used. It is important to underlined that the number of SISAR RPCs is remarkably lower (about 1/2 ,1/3 coefficients) than the one of ISD RPCs (78), being SISAR RPCs estimated considering the parsimony principle. In order to summarize the results of the comparison between RPCs generated by SISAR software (SISAR RPCs) and the RPCs supplied by vendors (ISD RPCs) the ratio R of the RMSE CP was introduced as reference index like listed in Tables 5.9, 5.10.

In this respect, which is a novelty of the implemented models, a particular care, both for rigorous orientation model and for RPCs model, was devoted into the estimable parameters selection. In fact not all the parameters included in a rigorous model might be well estimated from the available data; therefore, estimable parameters have to be automatically identified, whereas the non-estimable ones have to be properly constrained. As regards RPCs model until now the usual strategy for the selection of estimable coefficients was based on Tickhonov regularization. In this work a different innovative method for the estimable parameters selection was used, based on the Singular Value Decomposition (SVD) and QR decomposition.

One of the main application for satellite stereopair is the Digital Surface

Models (DSMs) generation. DSMs have large relevance in many enviromental and engeneering applications, such as topographic mapping, spatial and temporal change detection, feature extraction, data visualization, etc. Two DSMs were extracted and analysed: the first was obtained by the Cartosat-1 stereopairs and covers the CastelGandolfo Area (Lazio, central Italy) and the latter by the WorldView-1 stereopair covering the Augusta Area (Sicily). As regards Cartosat-1 DSM the height accuarcy is around 3 m in the one open area and around 4.5 m in the urban area; the accuracy increases after filtering (Tab.4.10). The WorldView-1 DSM accuracy was evaluated by sample comparisons, based on a significant number of Check Points with a mean 3D accuracy of 0.3-0.5 m, collected by kinematic GPS surveys and also on the DSM extracted from Quickbird stereopair (2.5 m average vertical accuracy); DSM accuracy is at 2 pixel level (Tab.4.13).

In the future the orientation model for new platforms (like WorldView-2 and Prism sensors) and the module to perform the image matching, based on two-stage dynamic programming technique will be implemented.

Table 4.5 Results in pixel of Level 1A single image orientation

RMSE CPs							
Sensor	**Area**	**SISAR**			**OrthoEngine**		
		n°GCPs	**E[pix]**	**N[pix]**	**n°CPs**	**E[pix]**	**N[pix]**
Level 1A images							
EROS A	Rome (R1)	17	1.33	1.39	17	1.14	1.06
	Rome (R2)	13	1.56	2.53	17	1.83	2.90
	Rome (R4)	13	1.26	1.28	17	1.33	1.61
QuickBird	Rome	17	0.72	0.77	13	0.89	0.84
	Augusta (*P001)	13	1.69	1.45	21	1.10	1.78
	Augusta (*P002)	13	1.28	1.25	13	1.24	1.41
	Salerno "joint"	17	0.78	1.25	21	1.07	0.97
Cartosat-1	Rome (bandA)	12	0.79	0.74	12	0.99	0.61
	Rome (bandF)	18	0.80	0.94	15	0.94	0.78
	CastelGandolfo (bandA)	21	0.98	1.04	21	0.78	0.83
	CastelGandolfo (bandF)	9	0.88	0.93	21	1.13	0.92
	Warsaw (bandA)	13	0.71	0.62	13	0.66	0.56
	Warsaw (bandF)	17	1.31	0.91	17	0.73	0.55
	Mausanne (bandA)	13	1.36	1.10	9	1.46	1.25
	Mausanne (bandF)	9	1.52	1.05	9	1.68	1.29
WorldView-1	Augusta (4505 R1C1-R2C1)	9	0.46	0.69	13	0.77	0.76
	Augusta (4545 R1C1-R2C1)	13	0.44	0.70	13	1.01	2.26

Table 4.7 Results in pixel of Level 1B single image orientation

RMSE CPs								
Sensor	**Area**	**Acquisition mode**	**SISAR**			**OrthoEngine**		
			n°GCPs	**E[pix]**	**N[pix]**	**n°GCPs**	**E[pix]**	**N[pix]**
Level 1B images								
Ikonos	Rome	Forward	10	0.86	0.72	10	0.97	0.74
GeoEye-1	Rome	Reverse	12	0.66	1.26	8	0.60	1.24
WorldView-1	Rome	Forward	8	0.52	1.0	10	1.14	0.74
QuickBird	Rome	Forward	10	1.08	0.60	10	0.98	0.76

Table 4.8 Results in meter of level 1A - level 1B stereopair orientation

RMSE CPs									
Sensor	**Area**	**GSD[m]**	**n°GCPs**	**SISAR**			**OrthoEngine**		
				E[m]	**N[m]**	**H[m]**	**E[m]**	**N[m]**	**H[m]**
Level 1A images									
EROS A	Rome (R1/R2)	3.10	21	2.50	2.81	5.94	3.17	4.40	6.97
QuickBird	Augusta (*P001/*P002)	0.77	21	0.50	0.84	0.86	0.78	1.07	1.33
Cartosat-1	Rome (bandA/bandF)	2.50	15	1.58	1.60	2.27	2.07	1.63	2.13
	CastelGandolfo (bandA/bandF)	2.50	21	1.79	1.57	5.12	2.26	2.04	1.32
	Warsaw (bandA/bandF)	2.50	13	1.83	1.59	2.18	1.64	1.09	2.01
	Mausanne (bandA/bandF)	2.50	9	2.03	2.65	2.95	2.44	3.85	4.02
WorldView-1	Augusta (4505/4545)	0.50	17	0.42	0.59	0.98	0.90	1.50	4.08
	Augusta (4045 R1C1- 4545 R1C1)	0.50	12	0.35	0.32	0.72	0.53	0.41	1.14
	Augusta (4045 R2C1- 4545 R2C1)	0.50	12	0.48	0.25	1.37	1.03	1.71	4.99
Level 1B images									
Ikonos	Bagnoli (Bagnoli3/Bagnoli4)	1.00	13	1.76	1.38	2.02	0.90	1.55	1.95

Table 5.9 Ratio R in all tested software using shift transformation

RMSE CP(ISD RPCs)/RMSE CP(SISAR RPCs)							
Shift transformation	Image	OrthoEngine		Erdas 8.6		SISAR	
		I	J	I	J	I	J
QuickBird	Rome	0.19	1.04	0.5	0.86	0.64	0.79
	Augusta (*P001)	0.88	0.76	0.68	0.78	0.82	0.70
	Augusta (*P002)	0.27	1.09	0.38	1.36	0.33	1.18
	Salerno (*P001)	2.25	1.49	2.10	1.28	2.90	1.15
	Salerno (*P002)	1.01	0.41	1.28	0.54	1.32	0.51
	Salerno (*P003)	1.12	0.22	1.01	0.32	0.96	0.23
	Average	0.96	0.84	1.01	0.85	1.16	0.76
Ikonos	Rome	0.87	0.92	0.89	0.75	0.89	0.89
	Bagnoli1	0.88	1.10	0.87	1.13	0.91	1.14
	Bagnoli2	0.84	1.11	0.81	1.06	0.83	1.12
	Bagnoli3	1.03	0.87	0.99	1.08	1.02	0.89
	Bagnoli4	1.02	0.83	0.97	0.94	1.00	0.81
	Average	0.93	0.97	0.9	0.99	0.93	0.97

Table 5.10 Ratio R in all tested software using affine transformation

RMSE CP(ISD RPCs)/RMSE CP(SISAR RPCs)							
Affine transformation	Image	OrthoEngine		Erdas 8.6		SISAR	
		I	J	I	J	I	J
QuickBird	Rome	0.17	0.81	0.75	0.95	0.75	0.94
	Augusta (*P001)	0.93	1.16	1.09	1.16	0.96	1.14
	Augusta (*P002)	0.46	1.03	1.26	1.14	1.17	1.11
	Salerno (*P001)	1.28	1.91	0.82	1.81	0.80	1.73
	Salerno (*P002)	0.95	0.82	1.14	0.79	1.12	0.77
	Salerno (*P003)	0.75	0.61	1.13	0.93	1.14	0.93
	Average	0.82	1.07	1.27	1.12	1.11	1.10
Ikonos	Rome	1.02	1.07	1.01	1.07	1.01	1.10
	Bagnoli1	1.25	1.11	1.27	1.11	1.28	1.11
	Bagnoli2	1.13	1.14	1.15	1.14	1.15	1.14
	Bagnoli3	1.05	1.09	1.06	1.23	1.06	1.23
	Bagnoli4	1.08	1.06	1.10	1.13	1.10	1.12
	Average	1.11	1.10	1.12	1.14	1.12	1.14

Table 4.10 Accuracy of Cartosat-1 height models checked by precise reference DSM/DEM (*open area, **urban area)

Image	Area	SZ[m]	bias[m]	SZ =f(α=inclination)[m]
CastelGandolfo	no filtered	2.88*	-0.06	2.71+0.41*tan α
	filtered	2.29*	0.30	2.26+0.17*tan α
	no filtered	4.67**	-0.58	3.95+1.64*tan α
	filtered	4.06**	-0.34	3.27+1.91*tan α

Table 4.13 Results of accuracy assessment

	Open Area North		
	Bias [m]	St Dev [m]	RMSE Z [m]
DPCOR/SISAR WorldView-1-CPs	-0.60	1.14	1.29
OrthoEngine WorldView-1-CPs	1.32	0.57	1.44
OrthoEngine QuickBird-CPs	0.27	0.60	0.66
OrthoEngine QuickBird-DPCOR/SISAR WorldView-1	1.39	2.25	2.64
OrthoEngine QuickBird-OrthoEngine WorldView-1	0.06	2.11	2.11
DPCOR/SISAR WorldView-1-OrthoEngine WorldView-1	-1.69	1.90	2.52
	City North		
	Bias [m]	St Dev [m]	RMSE Z [m]
DPCOR/SISAR WorldView-1-CPs	0.21	0.59	0.63
OrthoEngine WorldView-1-CPs	0.34	0.64	0.72
OrthoEngine QuickBird-CPs	2.91	0.77	3.01
OrthoEngine QuickBird-DPCOR/SISAR WorldView-1	2.65	1.90	3.26
OrthoEngine QuickBird-OrthoEngine WorldView-1	2.26	1.83	2.91
DPCOR/SISAR WorldView-1-OrthoEngine WorldView-1	-0.29	1.56	1.59

Appendix A

A.1 Rigourous model for single image (level 1A)

A.1.1 EROS A

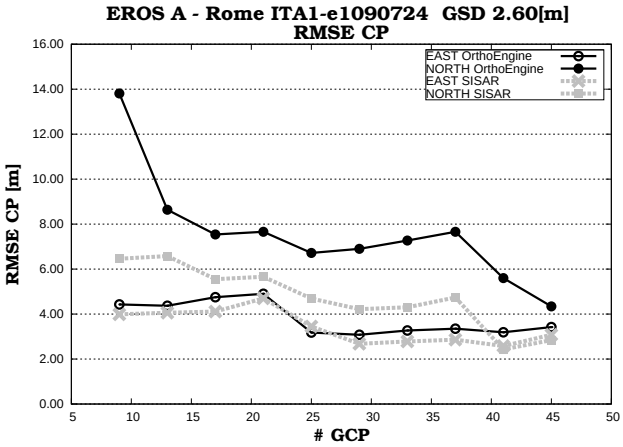


Figure A.1: Image accuracy vs. GCP number for EROS A (ITA1-e1090724)

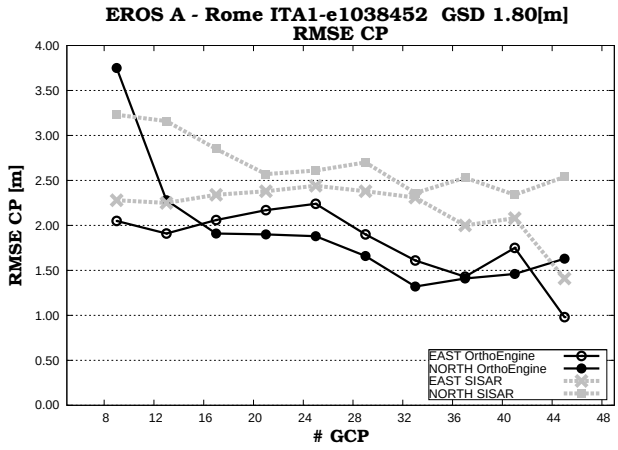


Figure A.2: Image accuracy vs. GCP number for EROS A (ITA1-e1038452)

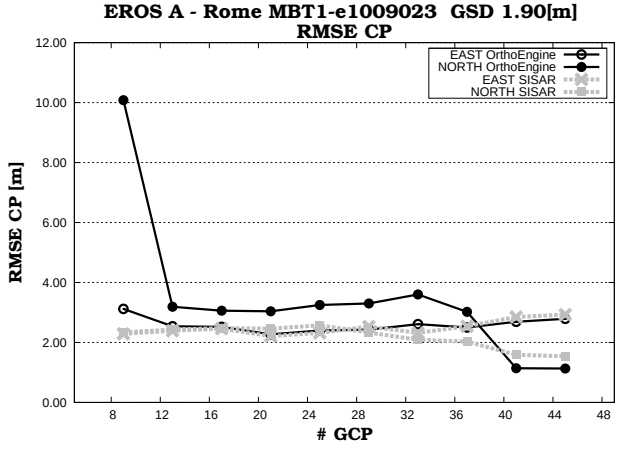


Figure A.3: Image accuracy vs. GCP number for EROS A (MBT1-e1009023)

A.1.2 QuickBird

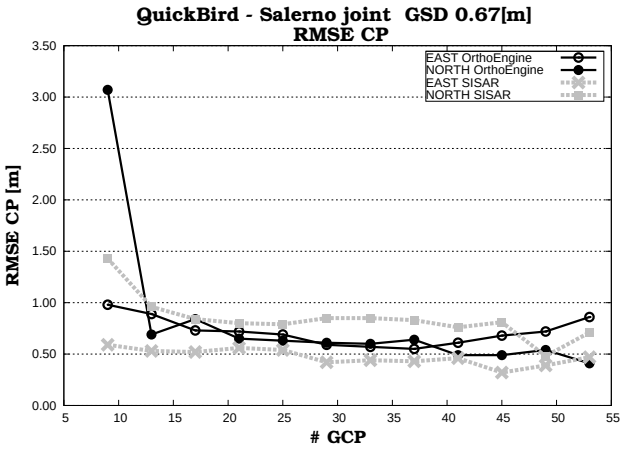


Figure A.4: Image accuracy vs. GCP number for QuickBird Salerno "joint"

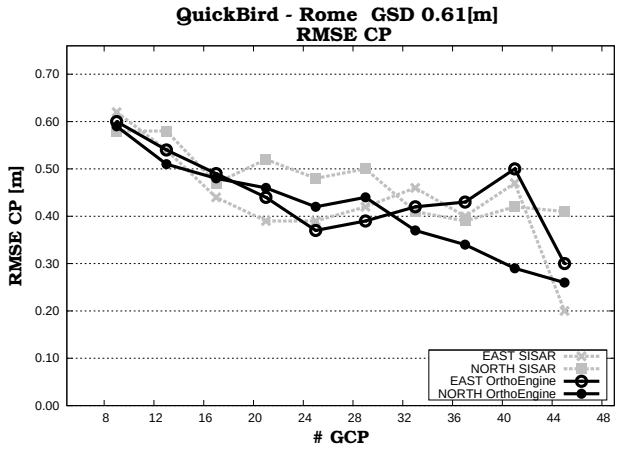


Figure A.5: Image accuracy vs. GCP number QuickBird image of Rome

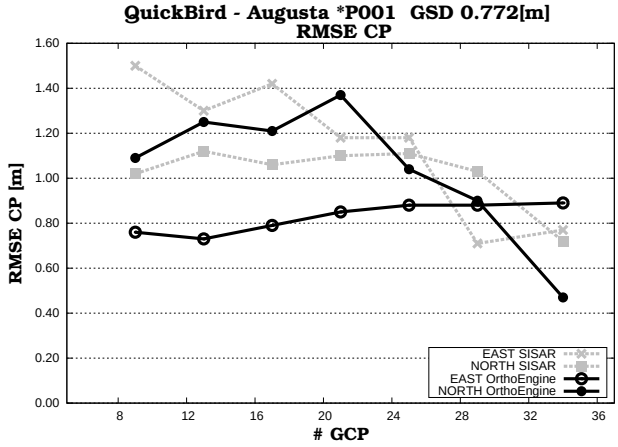


Figure A.6: Image accuracy vs. GCP number for QuickBird image of Augusta (*P001)

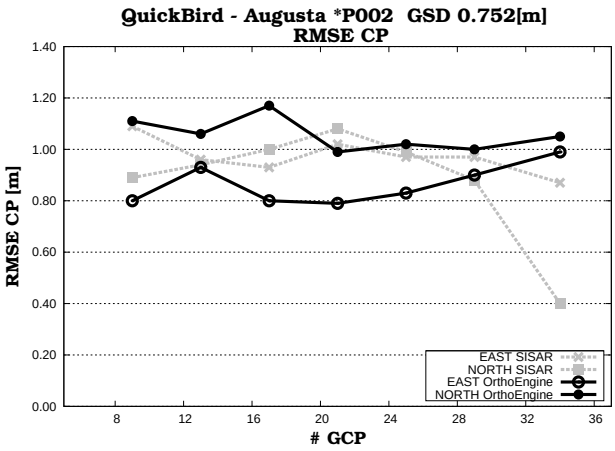


Figure A.7: Image accuracy vs. GCP number for QuickBird image of Augusta (*P002)

A.1.3 Cartosat-1

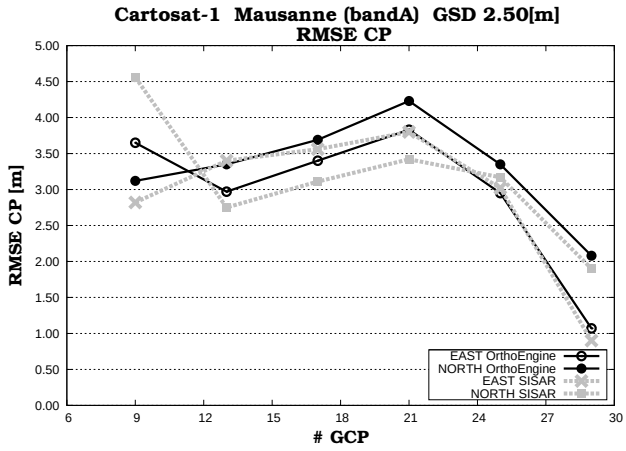


Figure A.8: Image accuracy vs. GCP number for Cartosat-1 Mausanne (bandA)

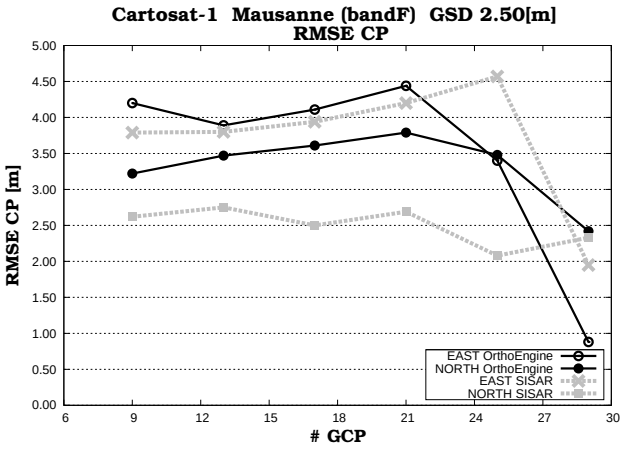


Figure A.9: Image accuracy vs. GCP number for Cartosat-1 Mausanne (bandF)

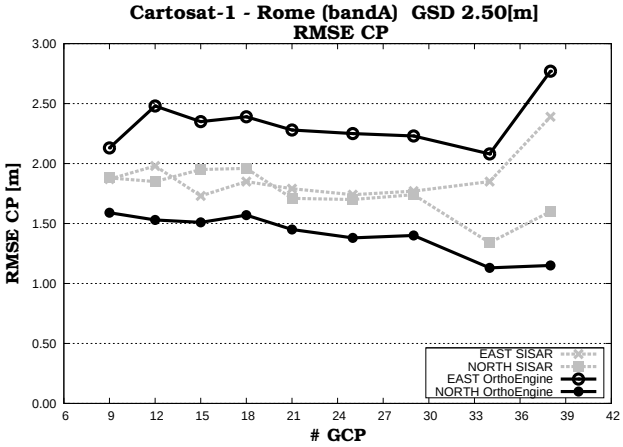


Figure A.10: Image accuracy vs. GCP number Cartosat-1 image of Rome (bandA)

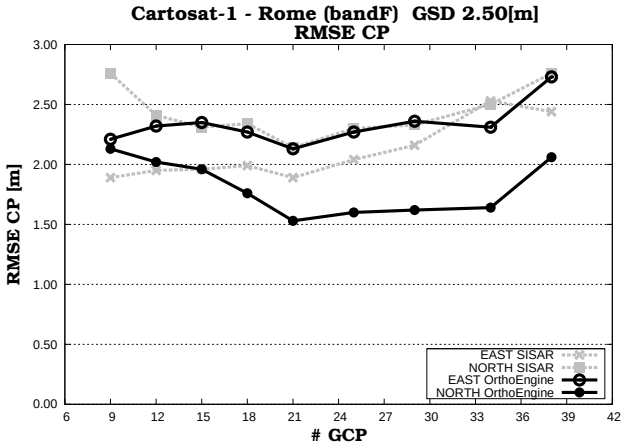


Figure A.11: Image accuracy vs. GCP number Cartosat-1 image of Rome (bandF)

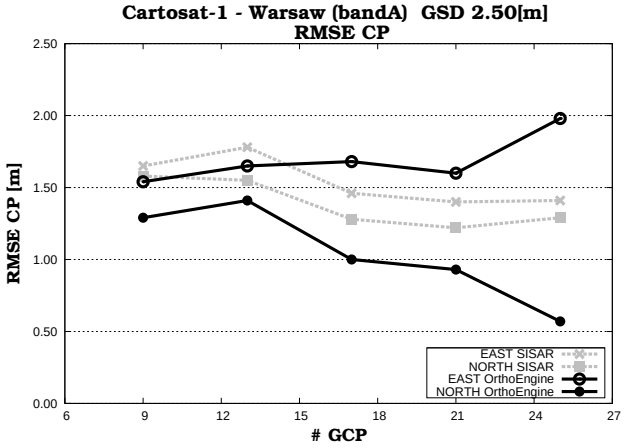


Figure A.12: Image accuracy vs. GCP number Cartosat-1 image of Warsaw (bandA)

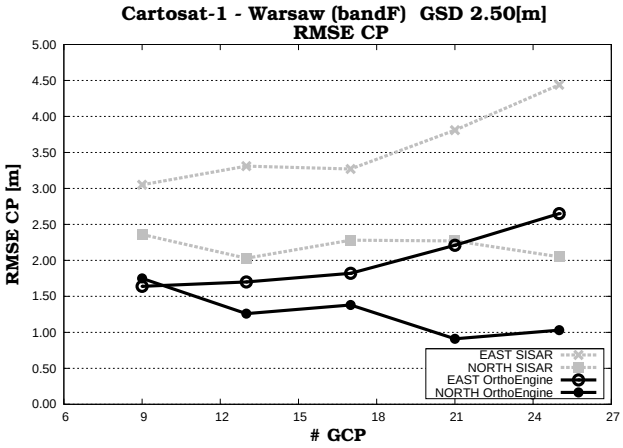


Figure A.13: Image accuracy vs. GCP number Cartosat-1 image of Warsaw (bandF)

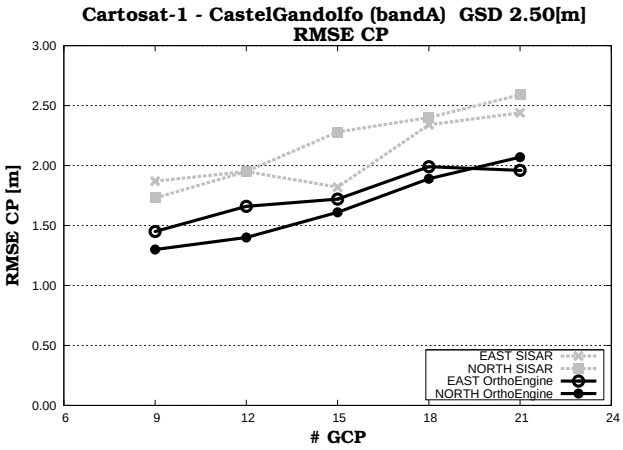


Figure A.14: Image accuracy vs. GCP number Cartosat-1 image of CastelGandolfo (bandA)

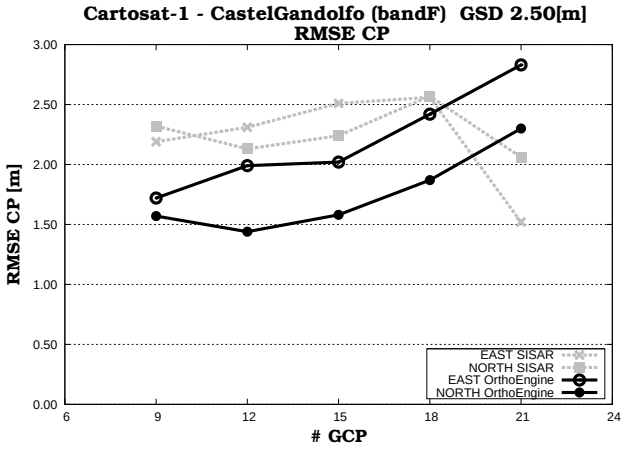


Figure A.15: Image accuracy vs. GCP number Cartosat-1 image of CastelGandolfo (bandF)

A.1.4 WorldView-1

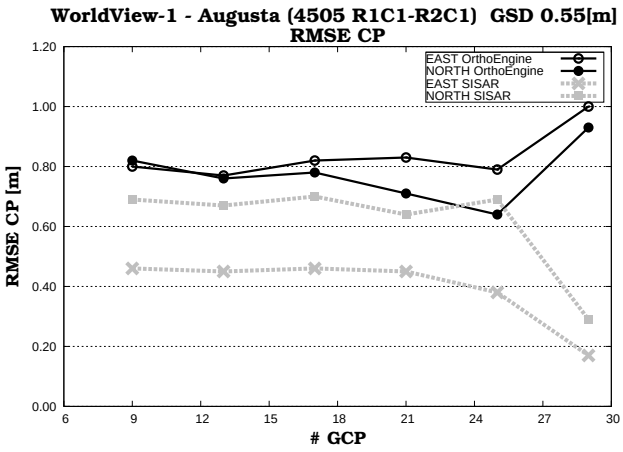


Figure A.16: Image accuracy vs. GCP number for WorldView-1 Augusta (4505 R1C1-R2C1)

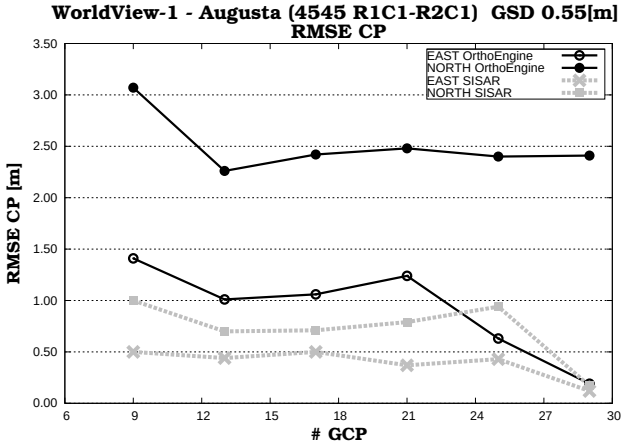


Figure A.17: Image accuracy vs. GCP number for WorldView-1 Augusta (4545 R1C1-R2C1)

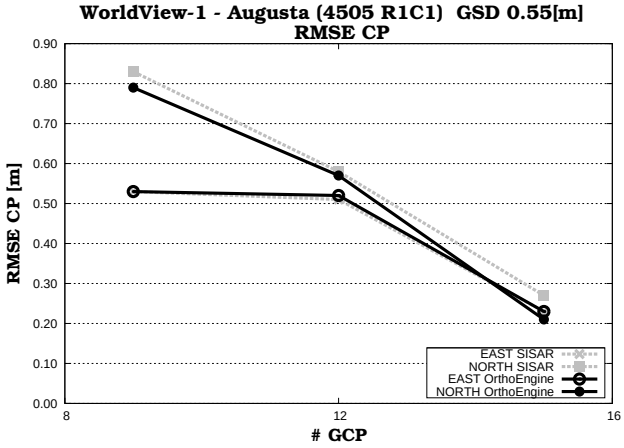


Figure A.18: Image accuracy vs. GCP number for WordlView-1 Augusta (4505 R1C1)

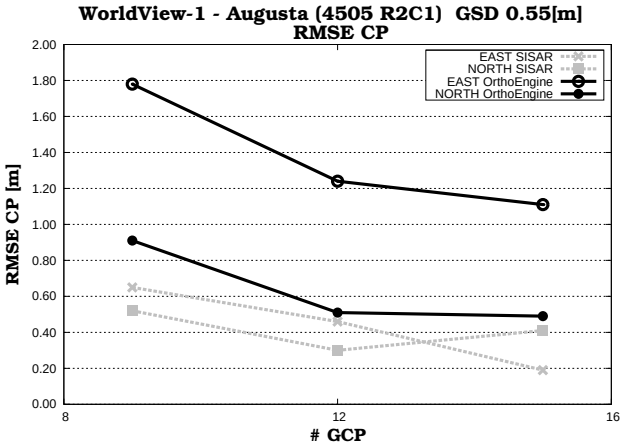


Figure A.19: Image accuracy vs. GCP number for WordlView-1 Augusta (4505 R2C1)

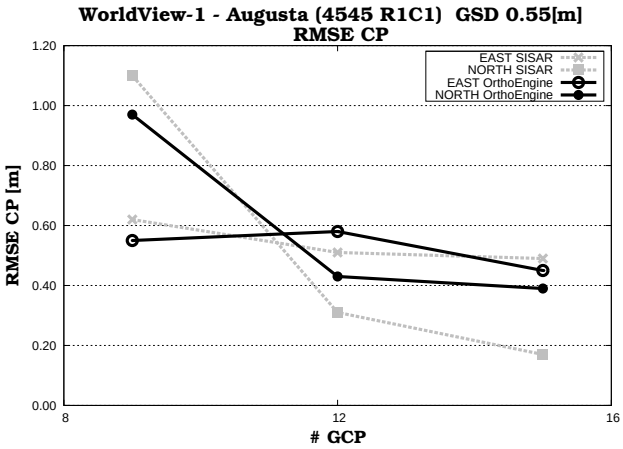


Figure A.20: Image accuracy vs. GCP number for WordlView-1 Augusta (4545 R1C1)

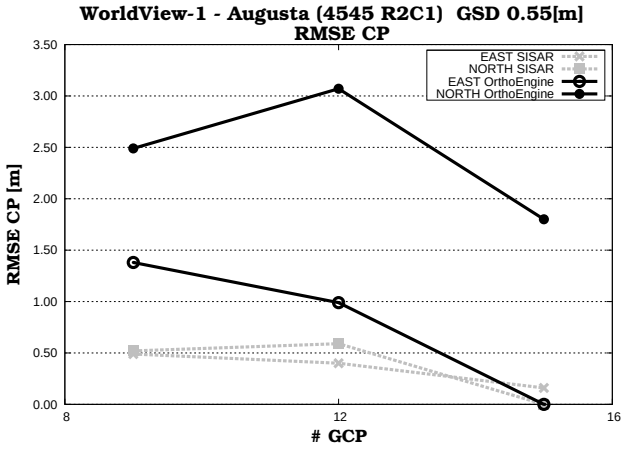


Figure A.21: Image accuracy vs. GCP number for WordlView-1 Augusta (4545 R2C1)

A.2 Rigourous model for single image (level 1B)

A.2.1 Ikonos

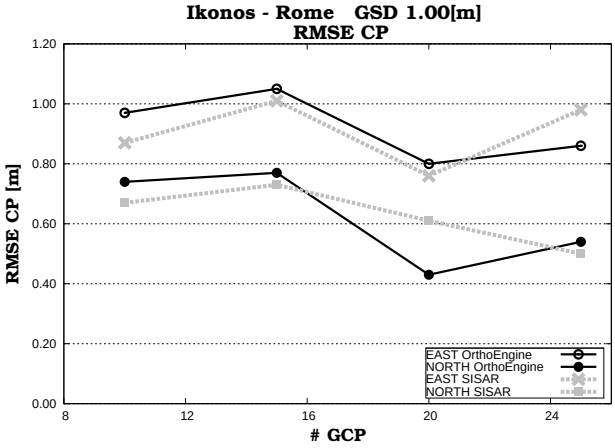


Figure A.22: Image accuracy vs. GCP number for Ikonos image of Rome

A.2.2 QuickBird

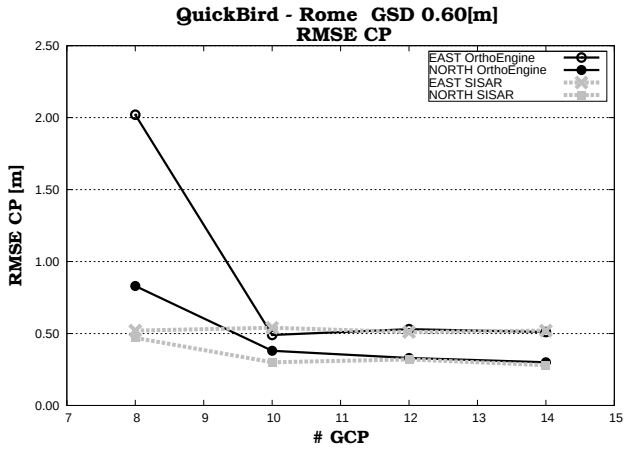


Figure A.23: Image accuracy vs. GCP number for QuickBird OrthoReady image of Rome

A.2.3 GeoEye-1

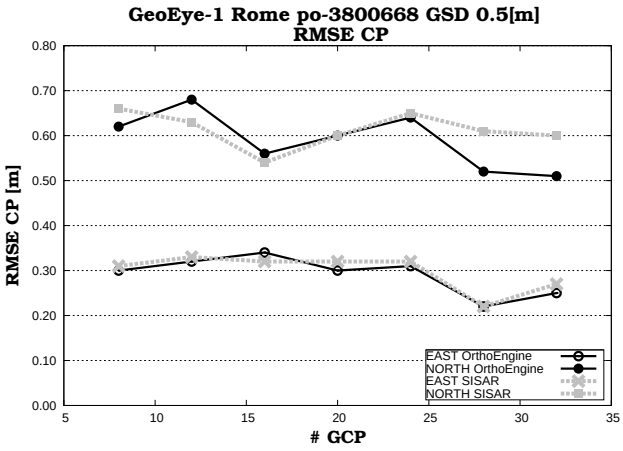


Figure A.24: Image accuracy vs. GCP number for GeoEye-1 image of Rome

A.2.4 WorldView-1

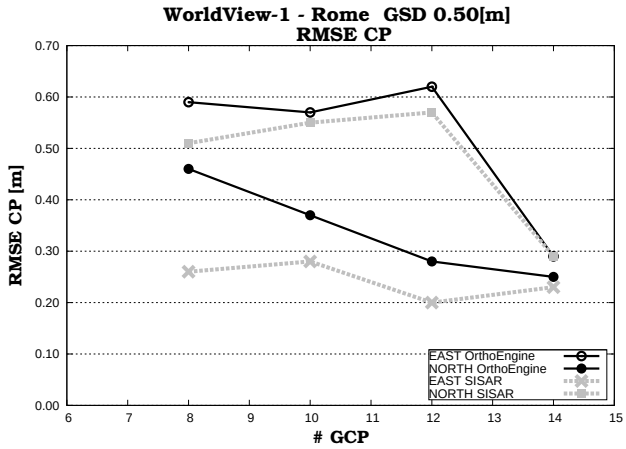


Figure A.25: Image accuracy vs. GCP number for WorldView-1 image of Rome

A.3 Rigorous model for stereopair (level 1A - level 1B)

A.3.1 EROS A

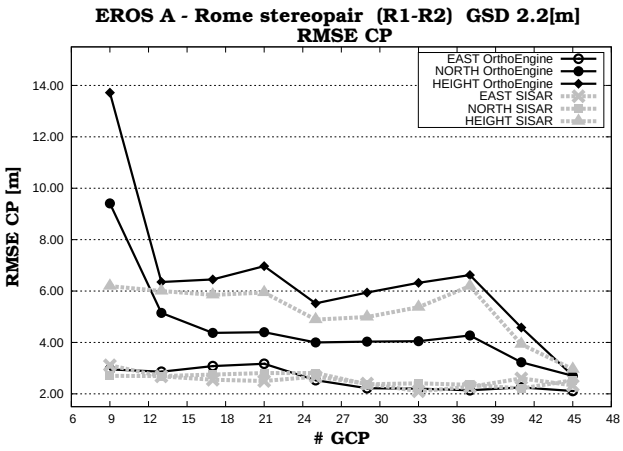


Figure A.26: Image accuracy vs. GCP number for EROS A stereopair of Rome (R1-R2)

A.3.2 QuickBird

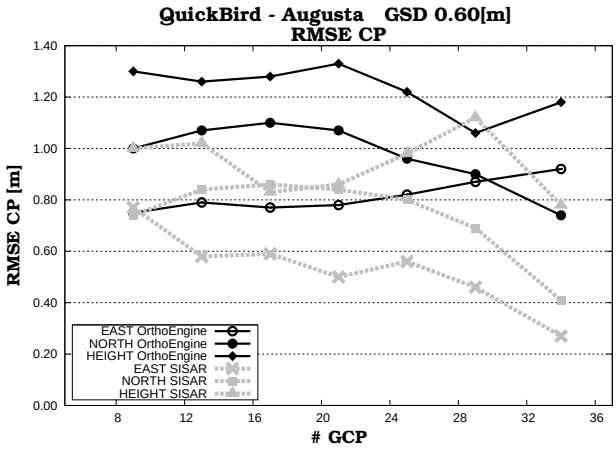


Figure A.27: Image accuracy vs. GCP number for QuickBird stereopair of Augusta

A.3.3 Cartosat-1

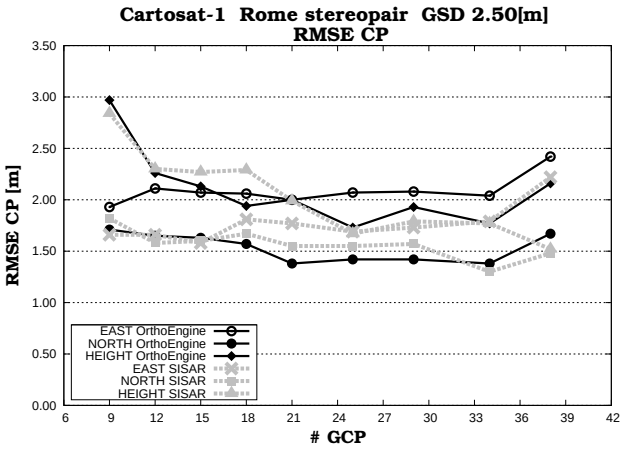


Figure A.28: Image accuracy vs. GCP number for Cartosat-1 stereopair of Rome

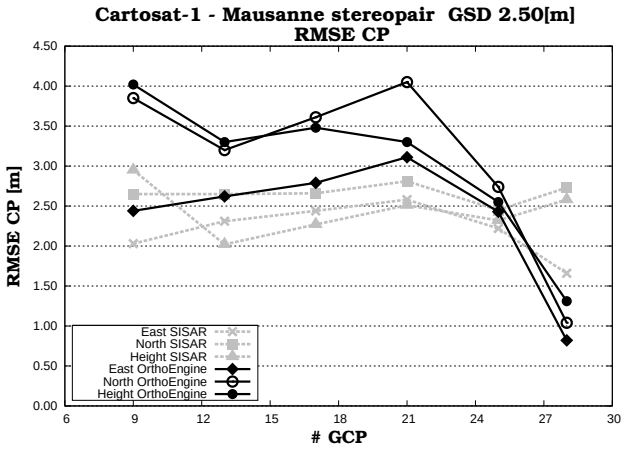


Figure A.29: Image accuracy vs. GCP number for Cartosat-1 Mausanne stereopair

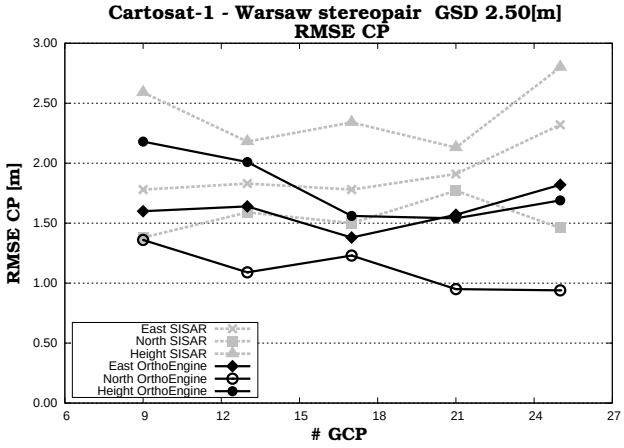


Figure A.30: Image accuracy vs. GCP number for Cartosat-1 Warsaw stereopair

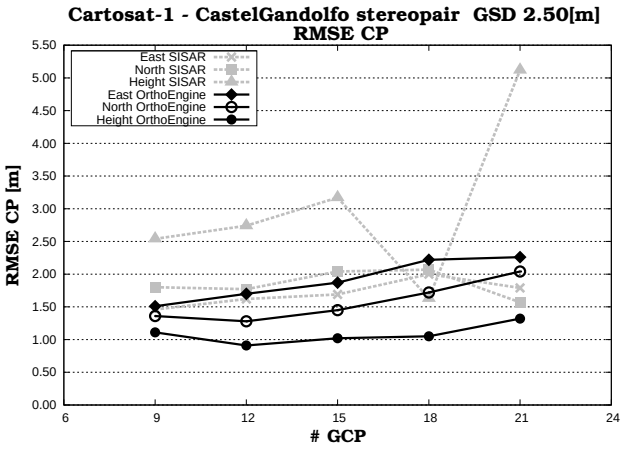


Figure A.31: Image accuracy vs. GCP number for Cartosat-1 CastelGandolfo stereopair

A.3.4 WorldView-1

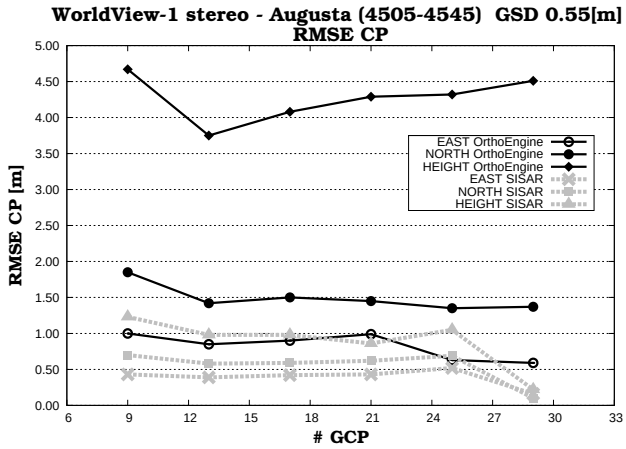


Figure A.32: Image accuracy vs. GCP number for WorldView-1 stereopair of Augusta (4505-4545)

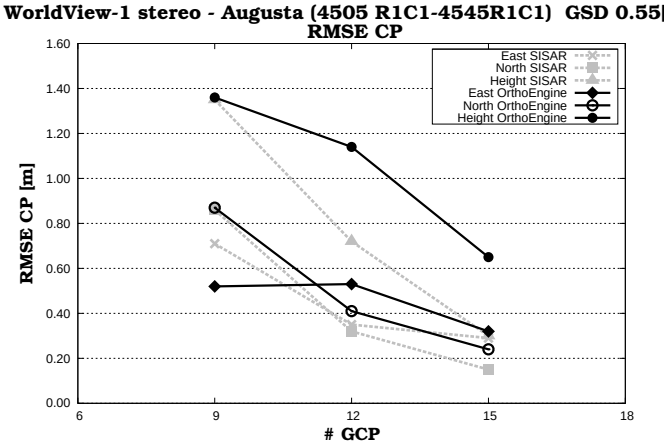


Figure A.33: Image accuracy vs. GCP number for WordlView-1 Augusta stereopair (4505 R1C1-4545 R1C1)

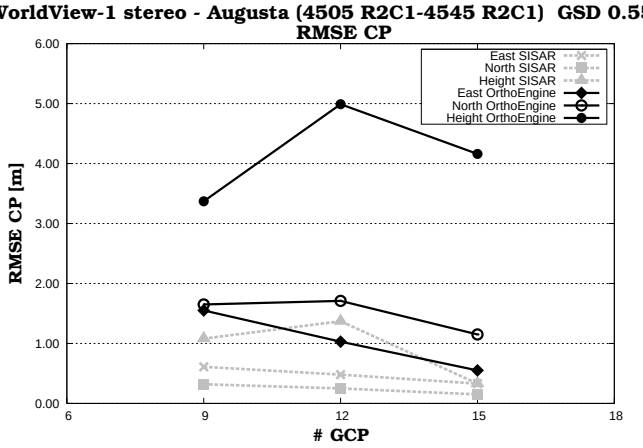


Figure A.34: Image accuracy vs. GCP number for WordlView-1 Augusta stereopair (4505 R2C1-4545 R2C1)

A.4 Rigorous model for stereopair (level 1B)

A.4.1 Ikonos

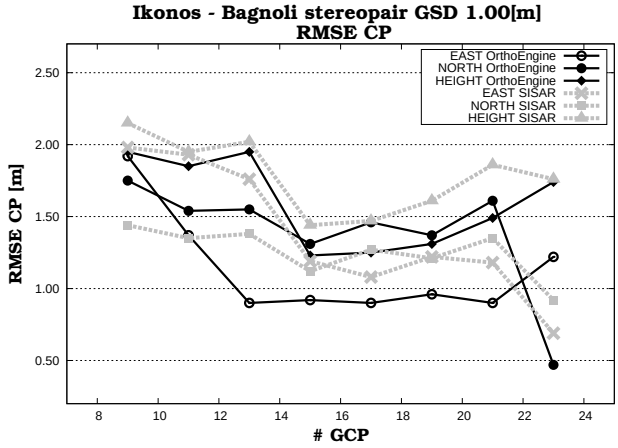


Figure A.35: Image accuracy vs. GCP number for Ikonos stereopair of Bagnoli3 - Bagnoli4

Appendix B

B.1 Results of RPCs application

B.1.1 Ikonos

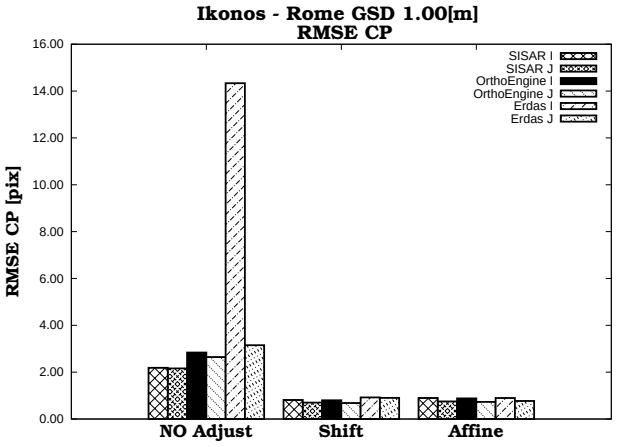


Figure B.1: Ikonos Rome - Image accuracy comparison with different correction techniques (5 GCP)

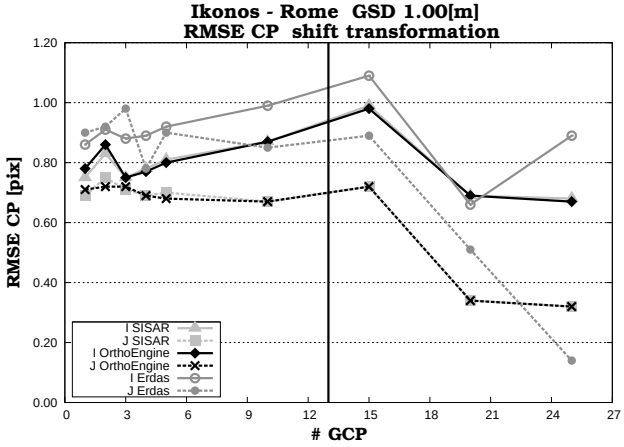


Figure B.2: Ikonos Rome - Image accuracy obtained with a shift transformation for the tested software

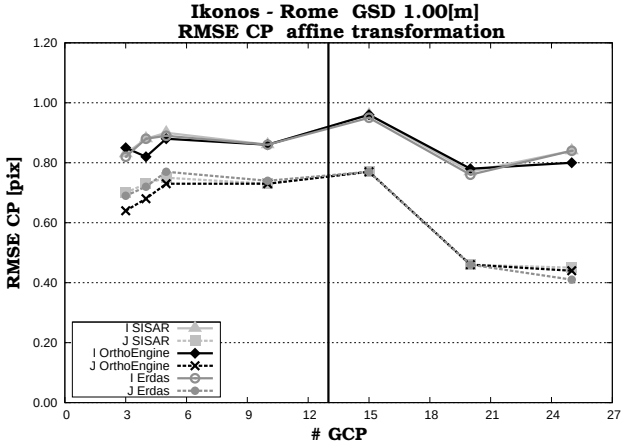


Figure B.3: Ikonos Rome - Image accuracy obtained with an affine transformation for the tested software

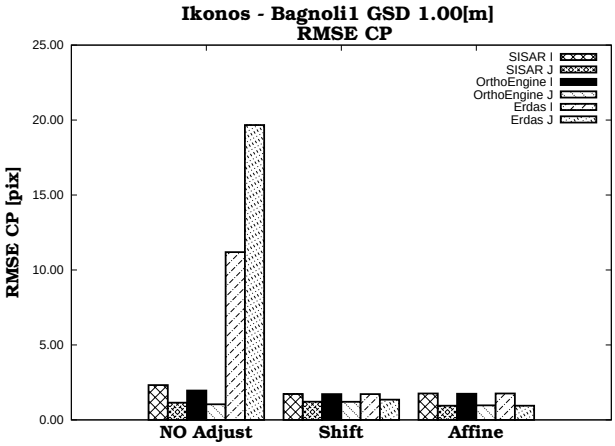


Figure B.4: Ikonos Bagnoli1 - Image accuracy comparison with different correction techniques (5 GCP)

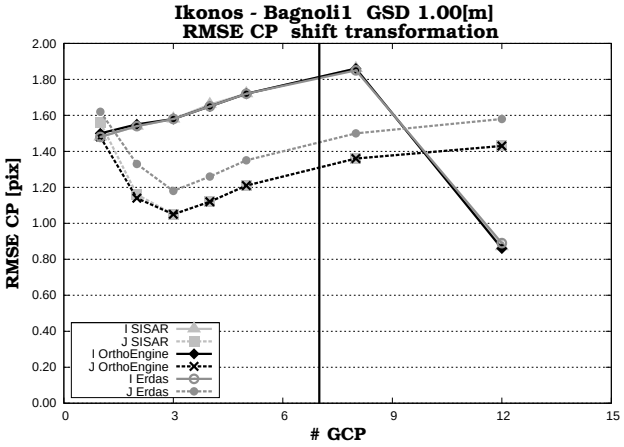


Figure B.5: Ikonos Bagnoli1 - Image accuracy obtained with a shift transformation for the tested software

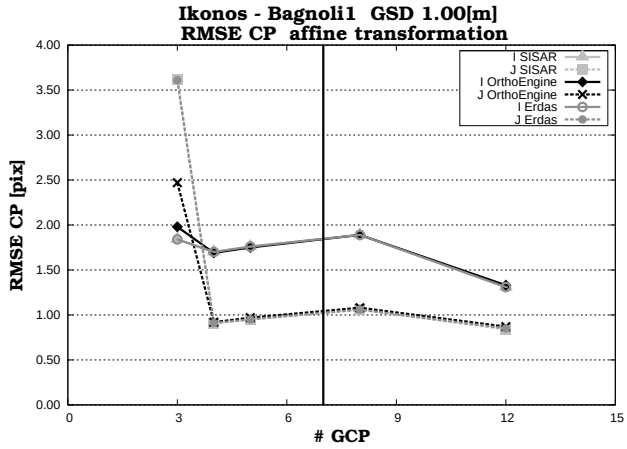


Figure B.6: Ikonos Bagnoli1 - Image accuracy obtained with an affine transformation for the tested software

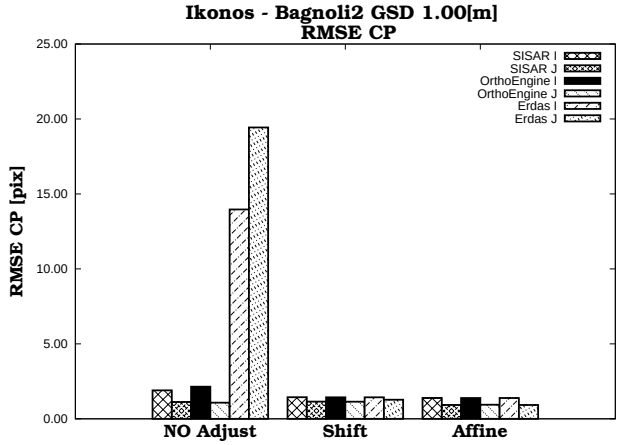


Figure B.7: Ikonos Bagnoli2 - Image accuracy comparison with different correction techniques (5 GCP)

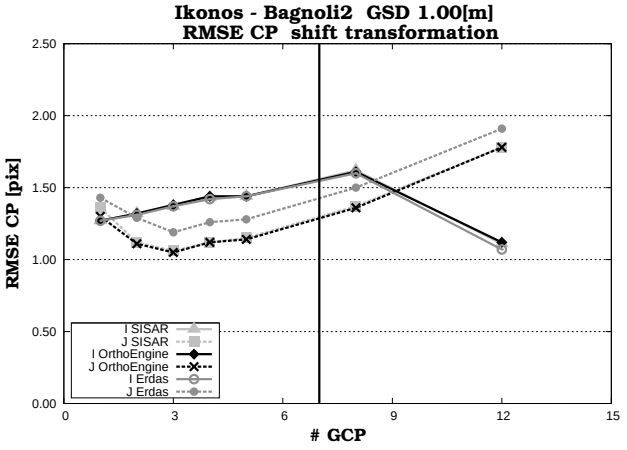


Figure B.8: Ikonos Bagnoli2 - Image accuracy obtained with a shift transformation for the tested software

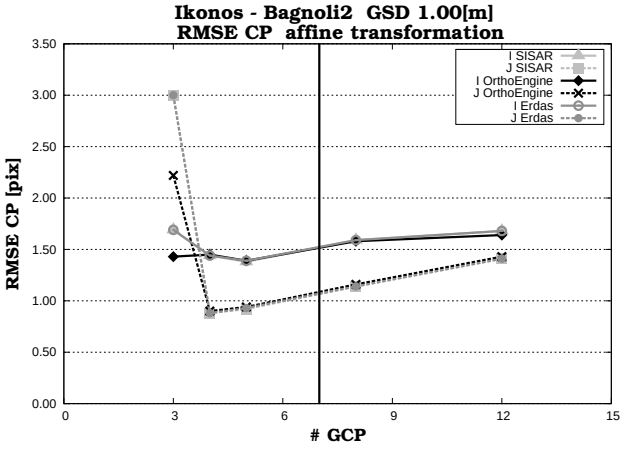


Figure B.9: Ikonos Bagnoli2 - Image accuracy obtained with an affine transformation for the tested software

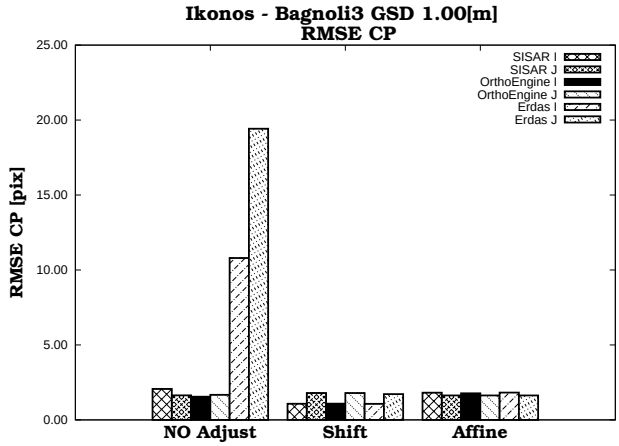


Figure B.10: Ikonos Bagnoli3 - Image accuracy comparison with different correction techniques (5 GCP)

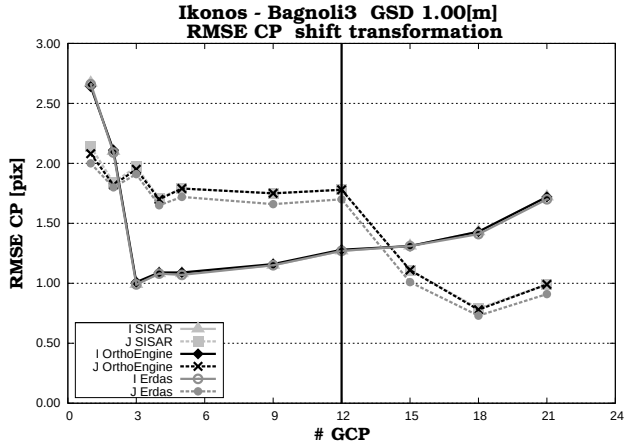


Figure B.11: Ikonos Bagnoli3 - Image accuracy obtained with a shift transformation for the tested software

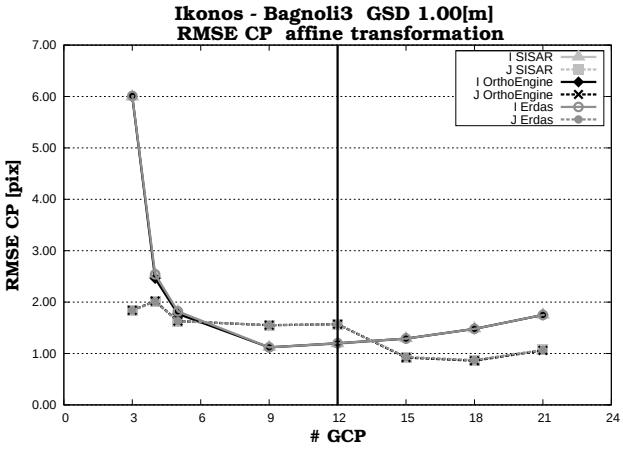


Figure B.12: Ikonos Bagnoli3 - Image accuracy obtained with an affine transformation for the tested software

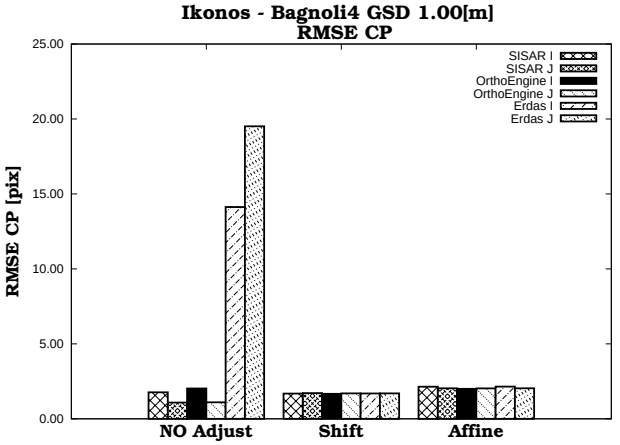


Figure B.13: Ikonos Bagnoli4 - Image accuracy comparison with different correction techniques (5 GCP)

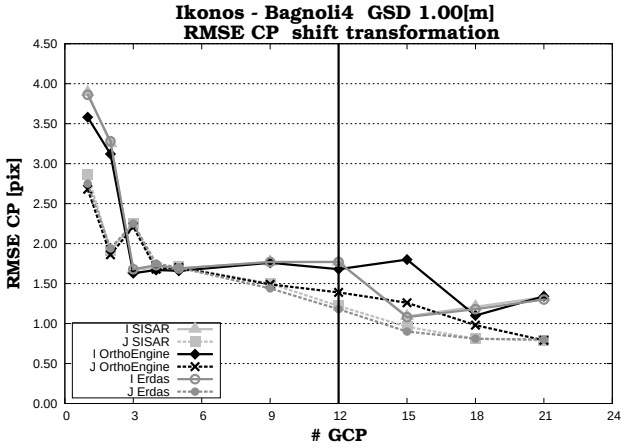


Figure B.14: Ikonos Bagnoli4 - Image accuracy obtained with a shift transformation for the tested software

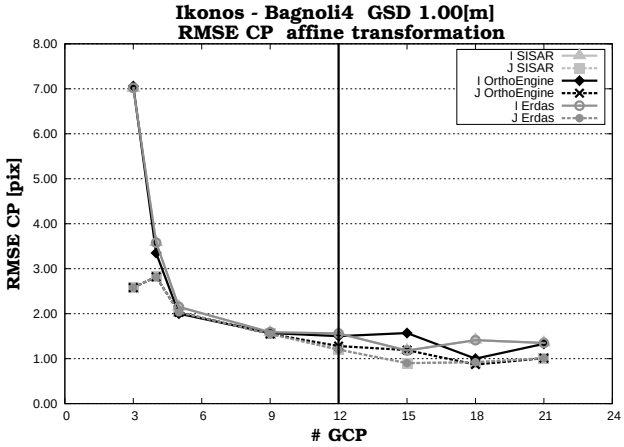


Figure B.15: Ikonos Bagnoli4 - Image accuracy obtained with an affine transformation for the tested software

B.1.2 QuickBird

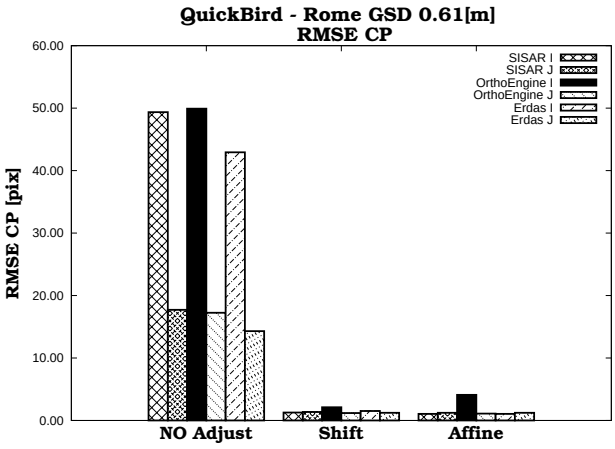


Figure B.16: QuickBird Rome - Image accuracy comparison with different correction techniques (5 GCP)

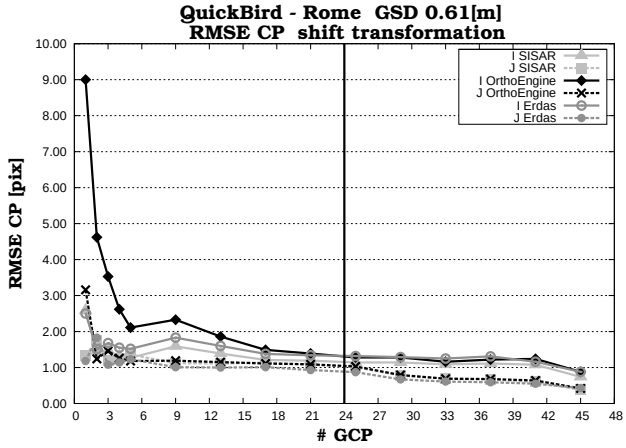


Figure B.17: QuickBird Rome - Image accuracy obtained with a shift transformation for the tested software

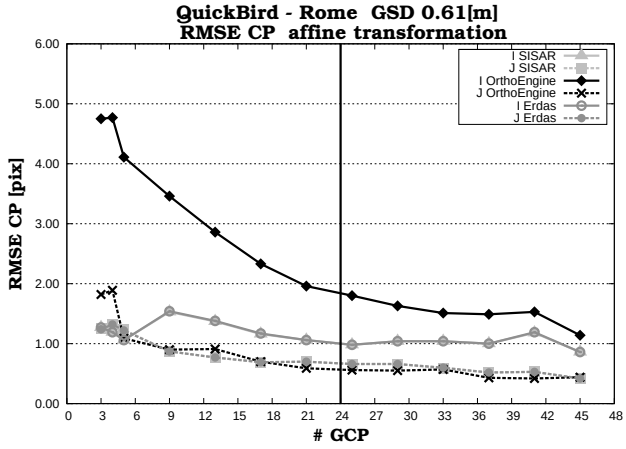


Figure B.18: QuickBird Rome - Image accuracy obtained with an affine transformation for the tested software

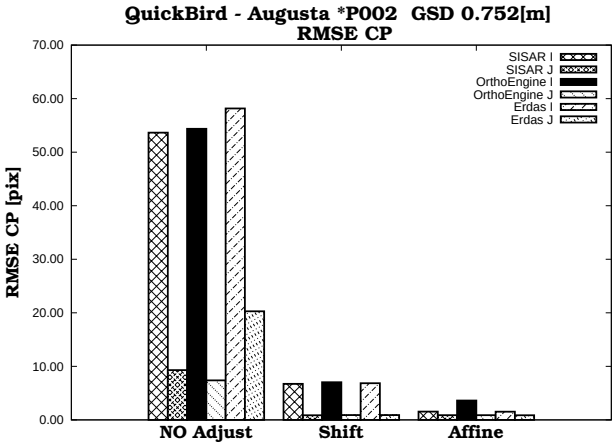


Figure B.19: QuickBird Augusta (*P002) - Image accuracy comparison with different correction techniques (5 GCP)

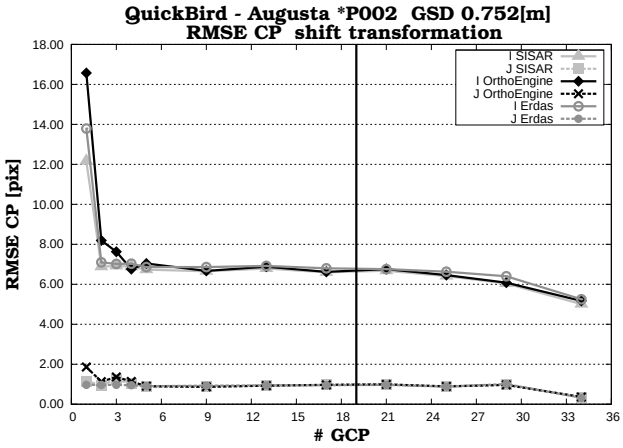


Figure B.20: QuickBird Augusta (*P002) - Image accuracy obtained with a shift transformation for the tested software

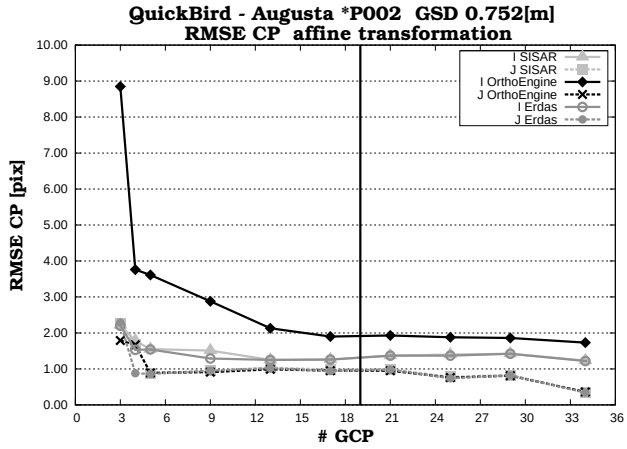


Figure B.21: QuickBird Augusta (*P002) - Image accuracy obtained with an affine transformation for the tested software

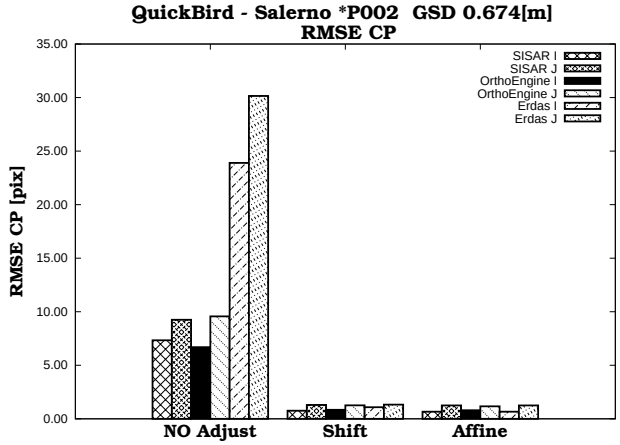


Figure B.22: QuickBird Salerno (*P002) - Image accuracy comparison with different correction techniques (5 GCP)

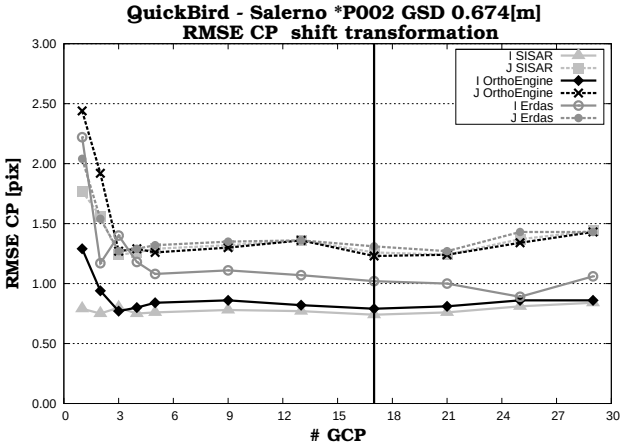


Figure B.23: QuickBird Salerno (*P002) - Image accuracy obtained with a shift transformation for the tested software

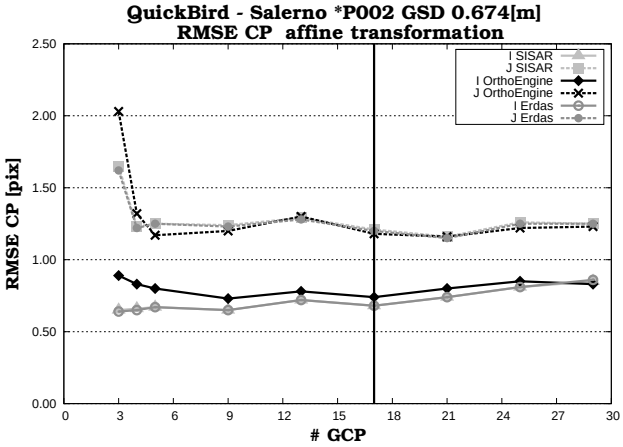


Figure B.24: QuickBird Salerno (*P002) - Image accuracy obtained with an affine transformation for the tested software

B.2 Comparison between ISD RPCs and SISAR RPCs in SISAR software

B.2.1 QuickBird

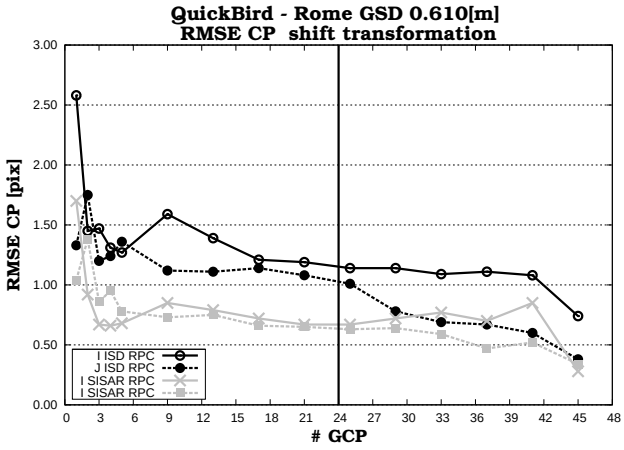


Figure B.25: QuickBird Rome - Image accuracy obtained with a shift transformation for SISAR software

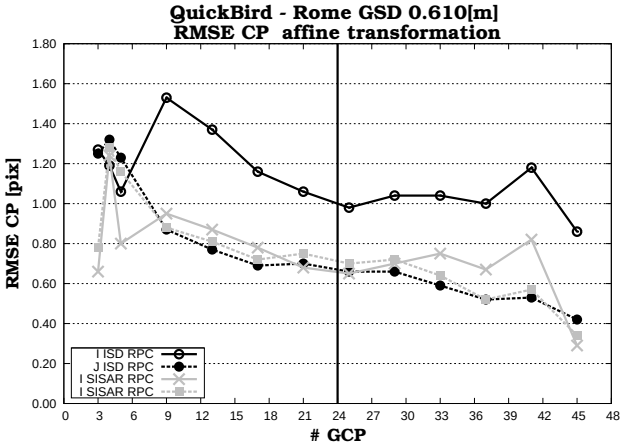


Figure B.26: QuickBird Rome - Image accuracy obtained with an affine transformation for SISAR software

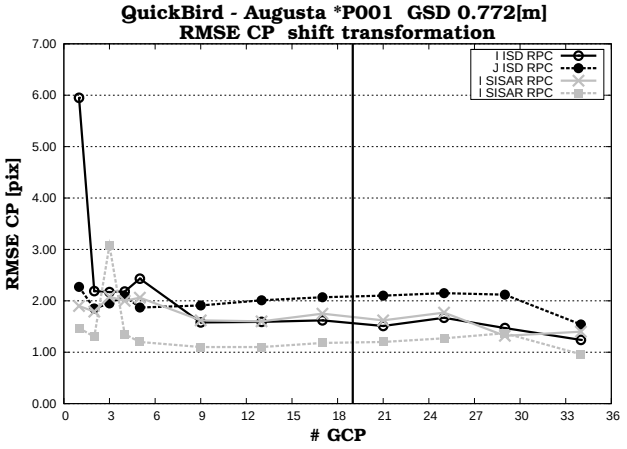


Figure B.27: QuickBird Augusta (*P001) - Image accuracy obtained with a shift transformation for SISAR software

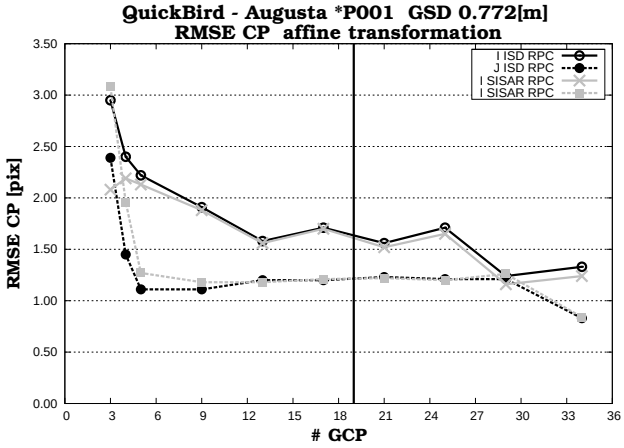


Figure B.28: QuickBird Augusta (*P001) - Image accuracy obtained with an affine transformation for SISAR software

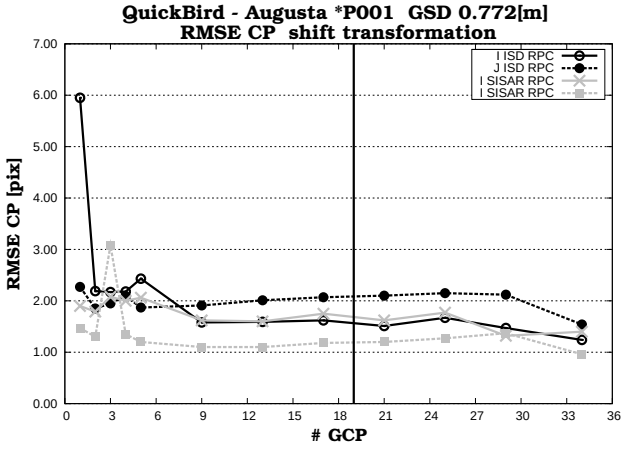


Figure B.29: QuickBird Augusta (*P002) - Image accuracy obtained with a shift transformation for SISAR software

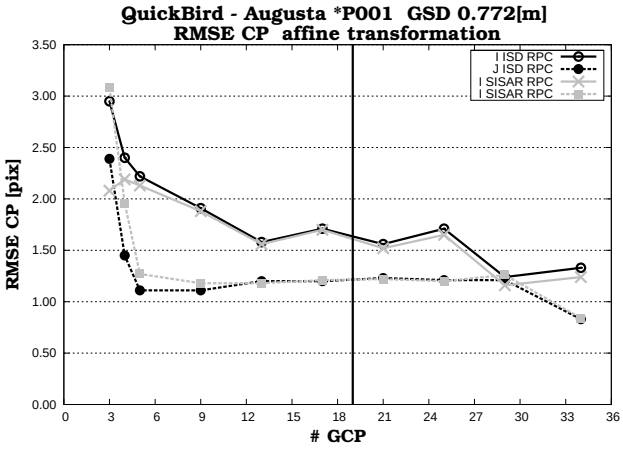


Figure B.30: QuickBird Augusta (*P002) - Image accuracy obtained with an affine transformation for SISAR software

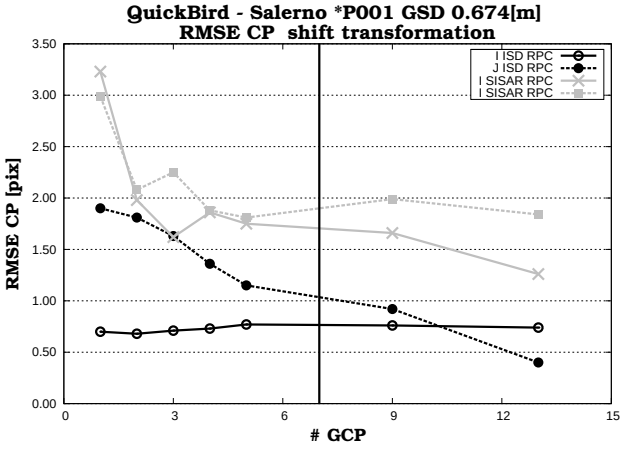


Figure B.31: QuickBird Salerno (*P001) - Image accuracy obtained with a shift transformation for SISAR software

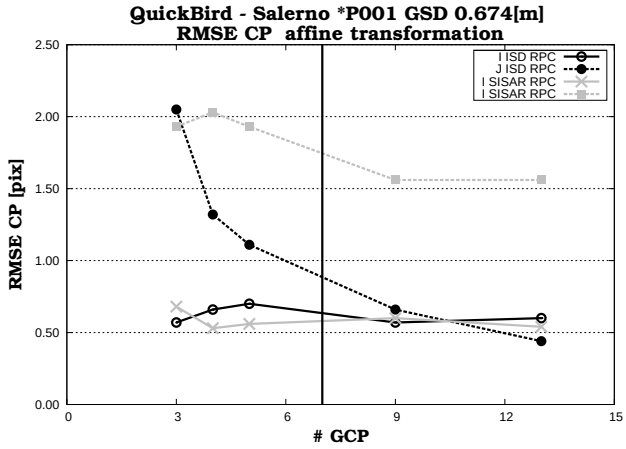


Figure B.32: QuickBird Salerno (*P001) - Image accuracy obtained with an affine transformation for SISAR software

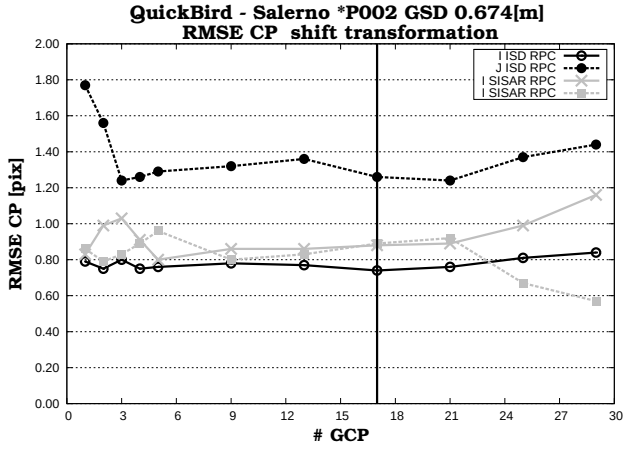


Figure B.33: QuickBird Salerno (*P002) - Image accuracy obtained with a shift transformation for SISAR software

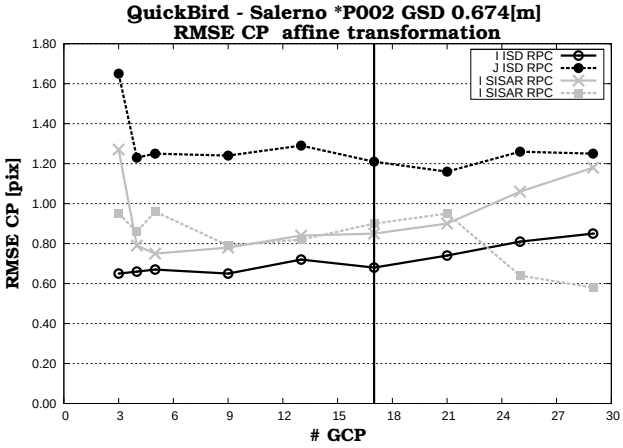


Figure B.34: QuickBird Salerno (*P002) - Image accuracy obtained with an affine transformation for SISAR software

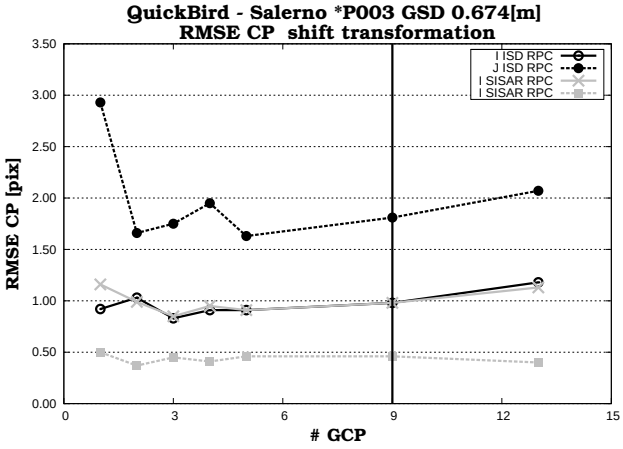


Figure B.35: QuickBird Salerno (*P003) - Image accuracy obtained with a shift transformation for SISAR software

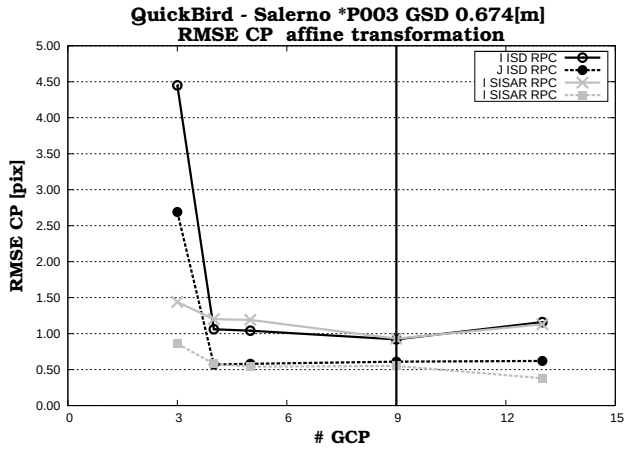


Figure B.36: QuickBird Salerno (*P003) - Image accuracy obtained with an affine transformation for SISAR software

B.2.2 Ikonos

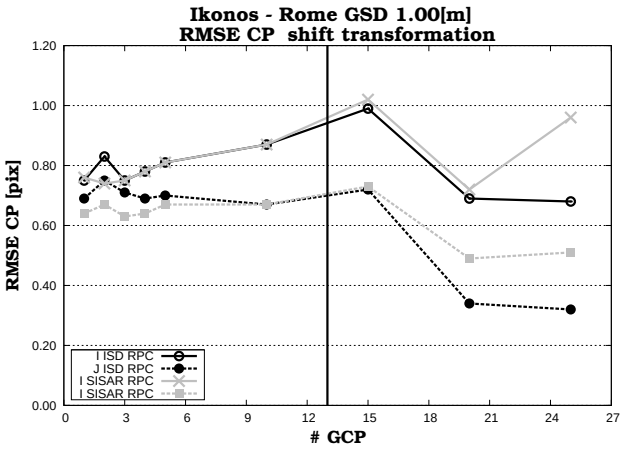


Figure B.37: Ikonos Rome - Image accuracy obtained with a shift transformation for SISAR software

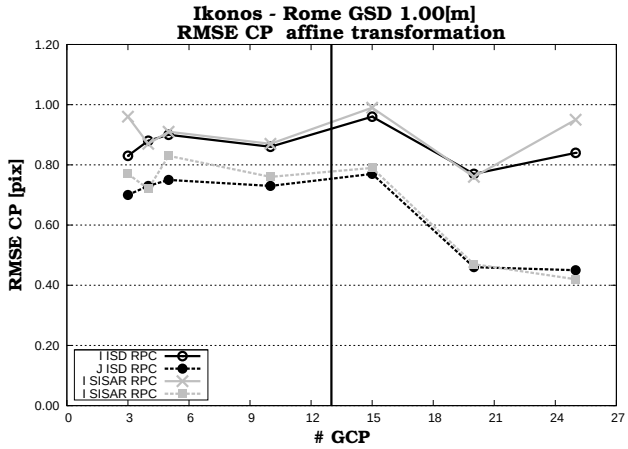


Figure B.38: Ikonos Rome - Image accuracy obtained with an affine transformation for SISAR software

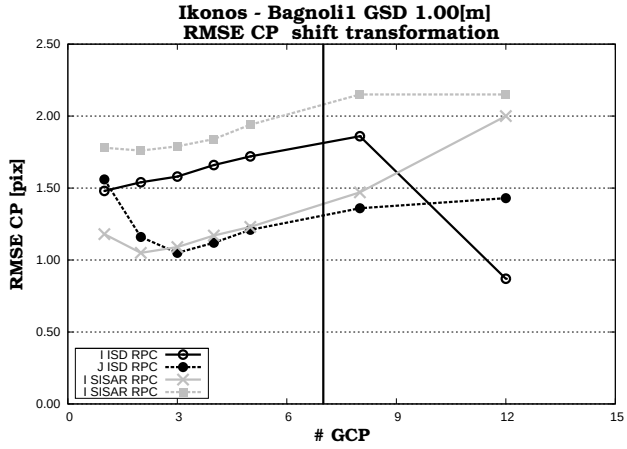


Figure B.39: Ikonos Bagnoli1 - Image accuracy obtained with a shift transformation for SISAR software

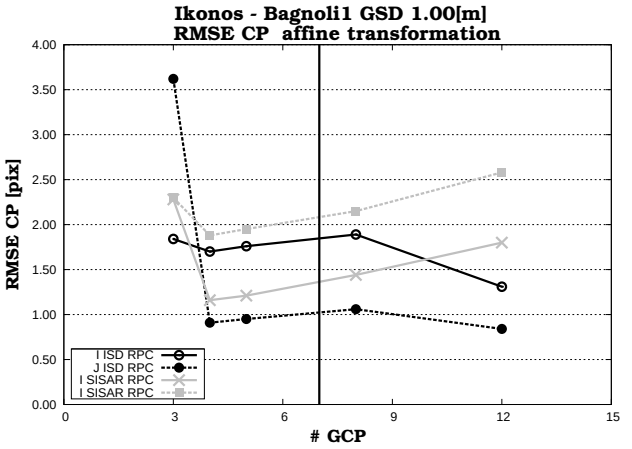


Figure B.40: Ikonos Bagnoli1 - Image accuracy obtained with an affine transformation for SISAR software

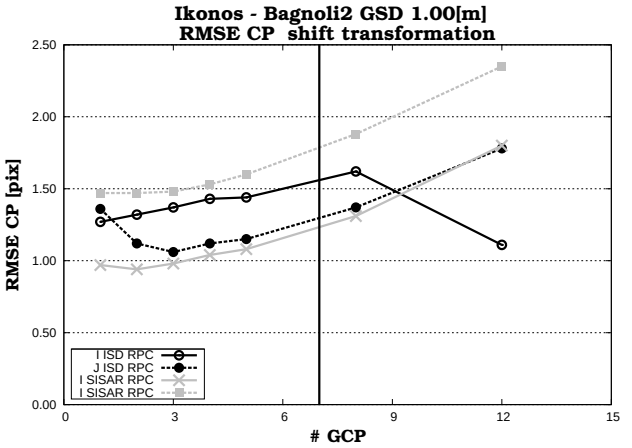


Figure B.41: Ikonos Bagnoli2 - Image accuracy obtained with a shift transformation for SISAR software

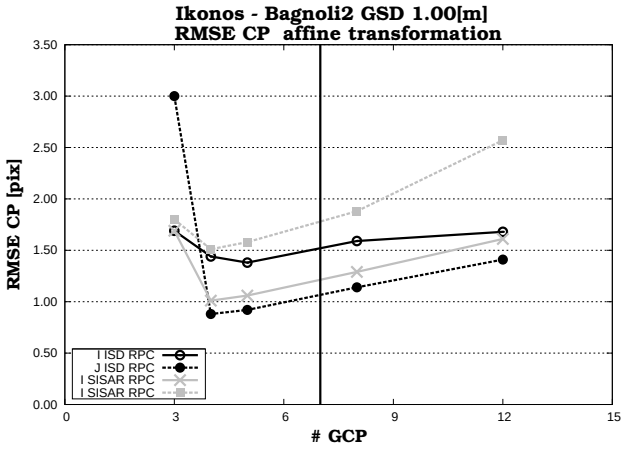


Figure B.42: Ikonos Bagnoli2 - Image accuracy obtained with an affine transformation for SISAR software

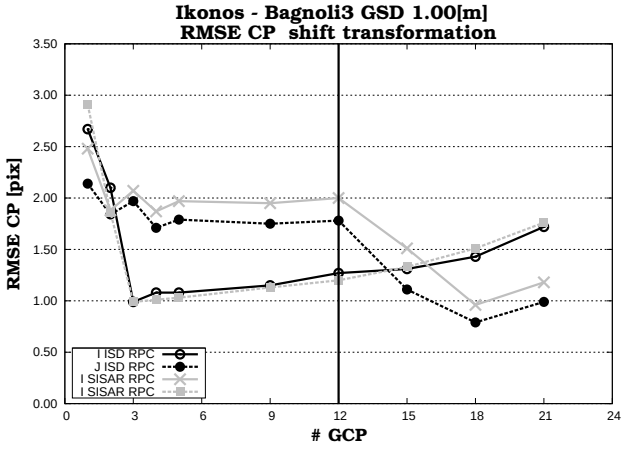


Figure B.43: Ikonos Bagnoli3 - Image accuracy obtained with a shift transformation for SISAR software

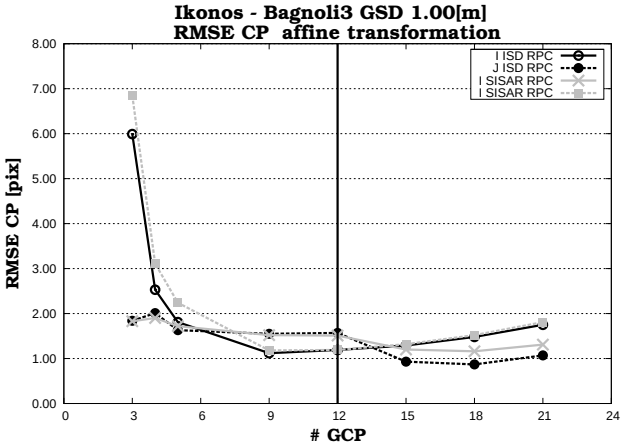


Figure B.44: Ikonos Bagnoli3 - Image accuracy obtained with an affine transformation for SISAR software

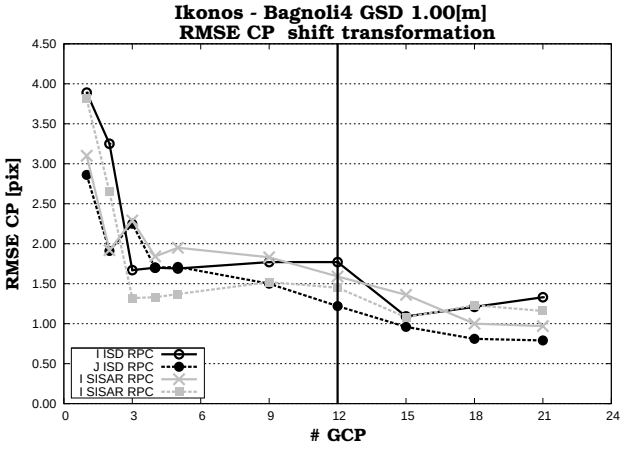


Figure B.45: Ikonos Bagnoli4 - Image accuracy obtained with a shift transformation for SISAR software

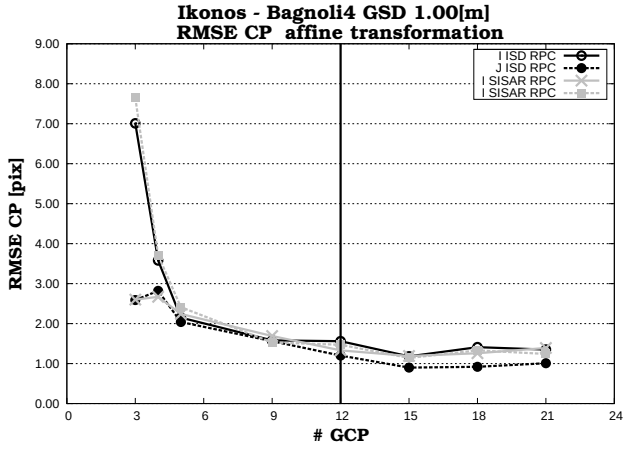


Figure B.46: Ikonos Bagnoli4 - Image accuracy obtained with an affine transformation for SISAR software

B.3 Comparison between ISD RPCs and SISAR RPCs in OrthoEngine software

B.3.1 QuickBird

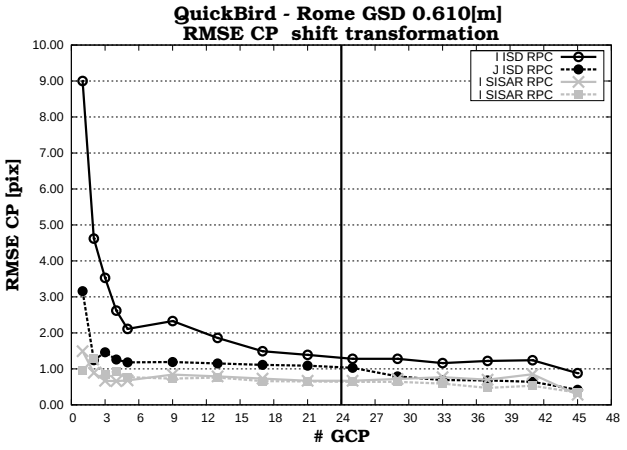


Figure B.47: QuickBird Rome - Image accuracy obtained with a shift transformation for OrthoEngine software

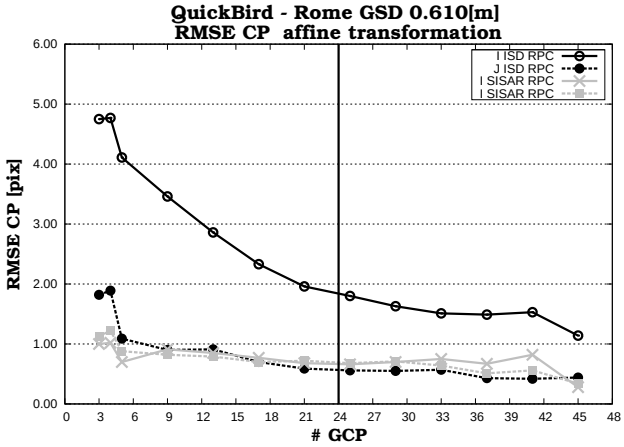


Figure B.48: QuickBird Rome - Image accuracy obtained with an affine transformation for OrthoEngine software

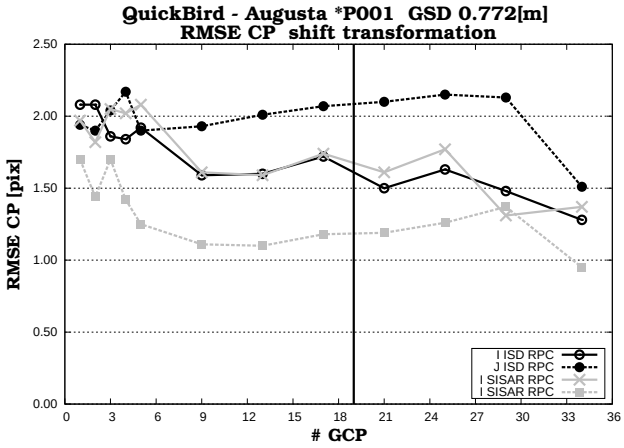


Figure B.49: QuickBird Augusta (*P001) - Image accuracy obtained with a shift transformation for OrthoEngine software

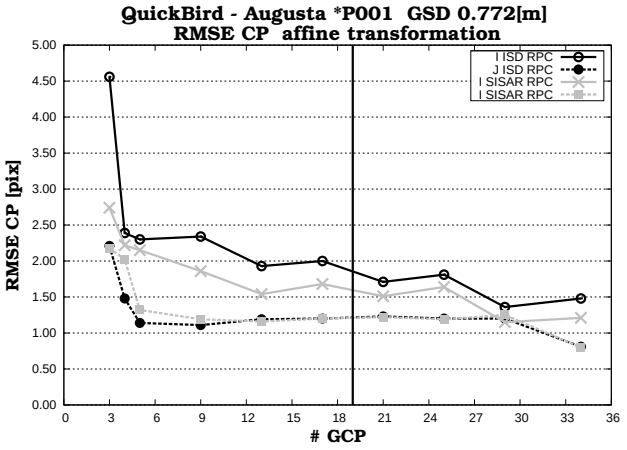


Figure B.50: QuickBird Augusta (*P001) - Image accuracy obtained with an affine transformation for OrthoEngine software

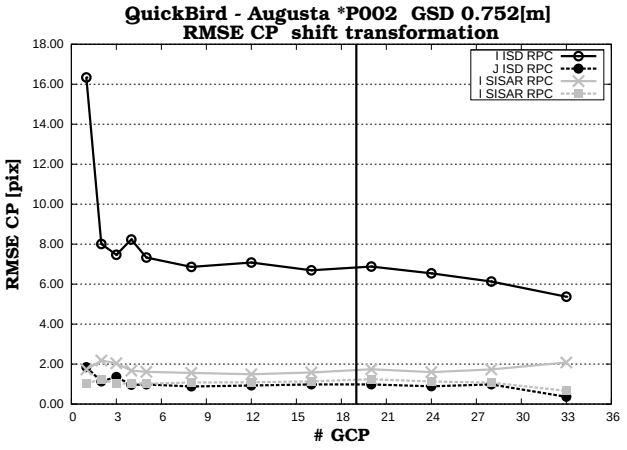


Figure B.51: QuickBird Augusta (*P002) - Image accuracy obtained with a shift transformation for OrthoEngine software

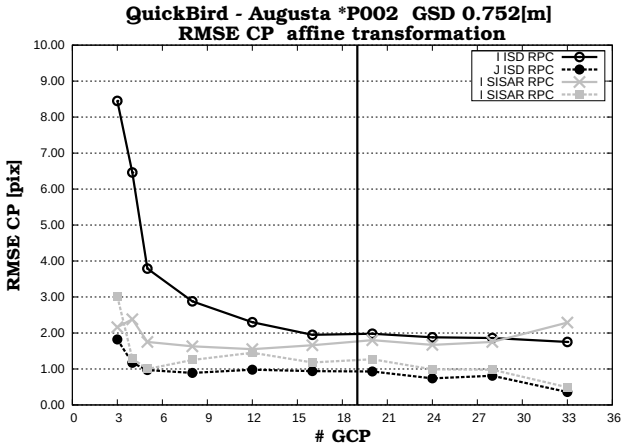


Figure B.52: QuickBird Augusta (*P002) - Image accuracy obtained with an affine transformation for OrthoEngine software

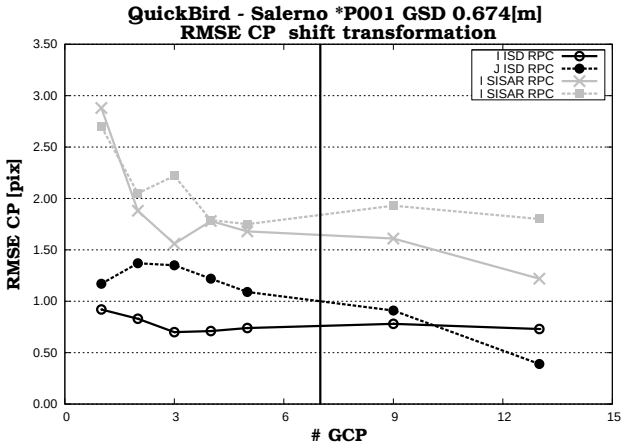


Figure B.53: QuickBird Salerno (*P001) - Image accuracy obtained with a shift transformation for OrthoEngine software

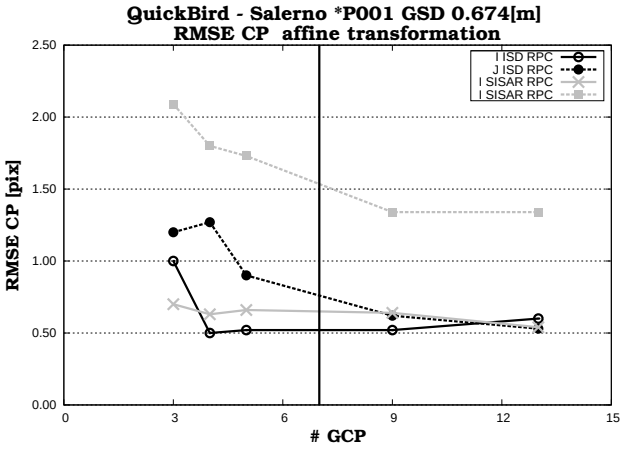


Figure B.54: QuickBird Salerno (*P001) - Image accuracy obtained with an affine transformation for OrthoEngine software

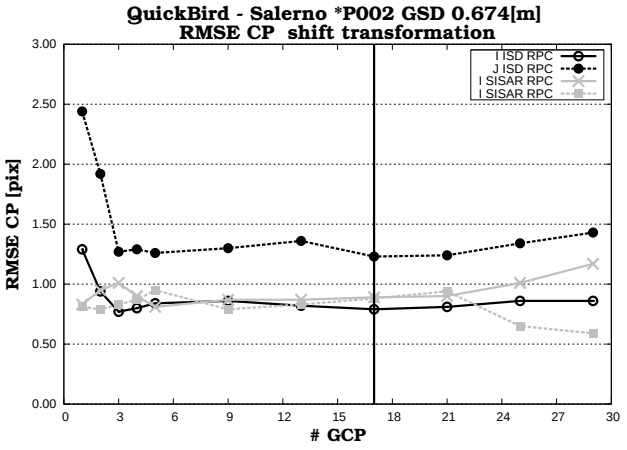


Figure B.55: QuickBird Salerno (*P002) - Image accuracy obtained with a shift transformation for OrthoEngine software

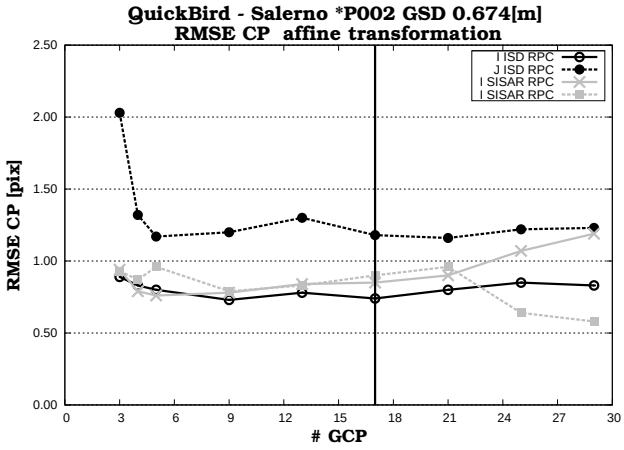


Figure B.56: QuickBird Salerno (*P002) - Image accuracy obtained with an affine transformation for OrthoEngine software

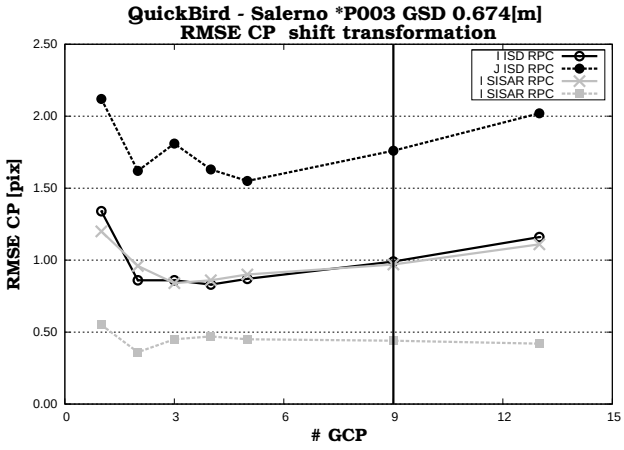


Figure B.57: QuickBird Salerno (*P003) - Image accuracy obtained with a shift transformation for OrthoEngine software

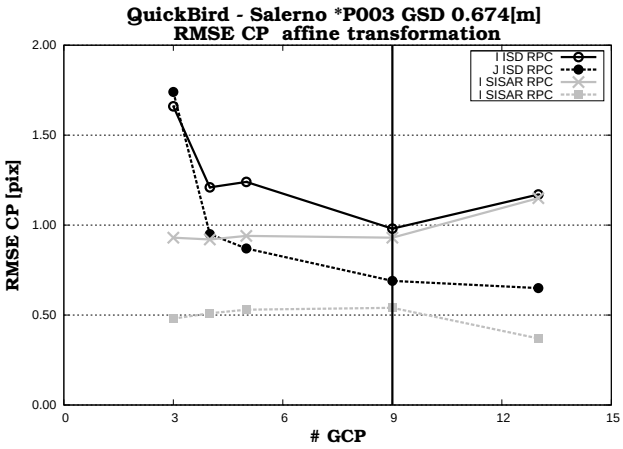


Figure B.58: QuickBird Salerno (*P003) - Image accuracy obtained with an affine transformation for OrthoEngine software

B.3.2 Ikonos

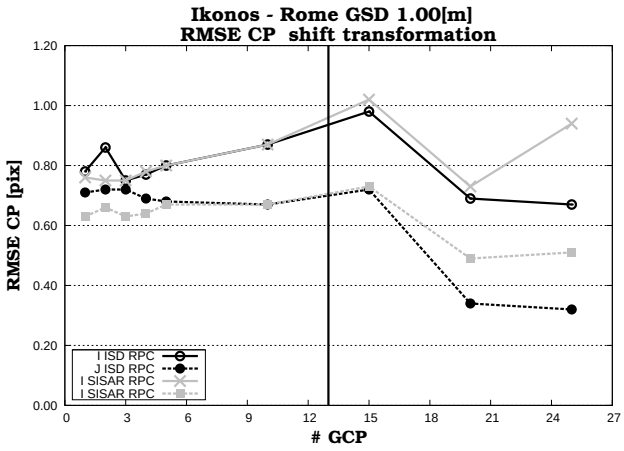


Figure B.59: Ikonos Rome - Image accuracy obtained with a shift transformation for OrthoEngine software

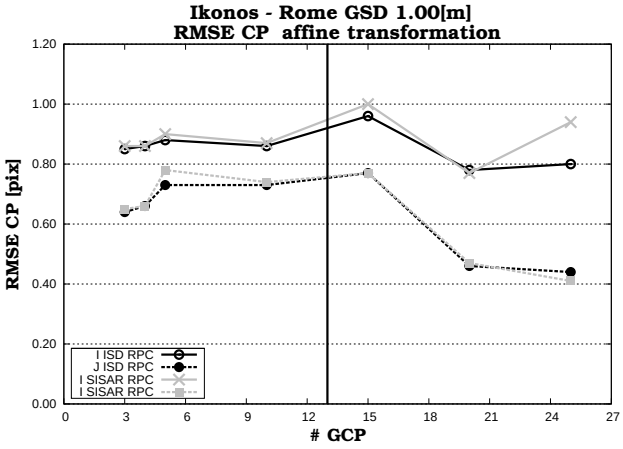


Figure B.60: Ikonos Rome - Image accuracy obtained with a affine transformation for OrthoEngine software

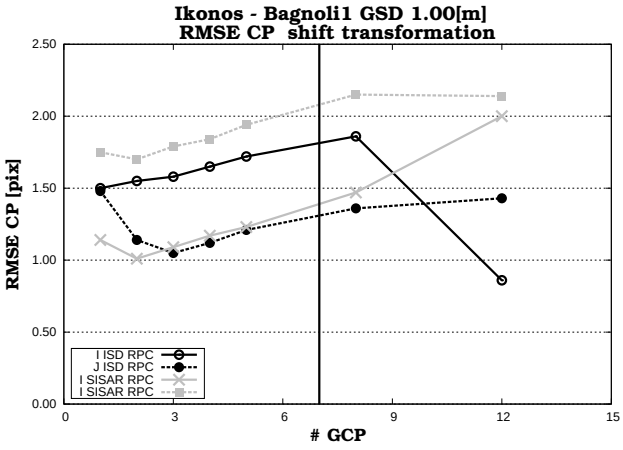


Figure B.61: Ikonos Bagnoli1 - Image accuracy obtained with a shift transformation for OrthoEngine software

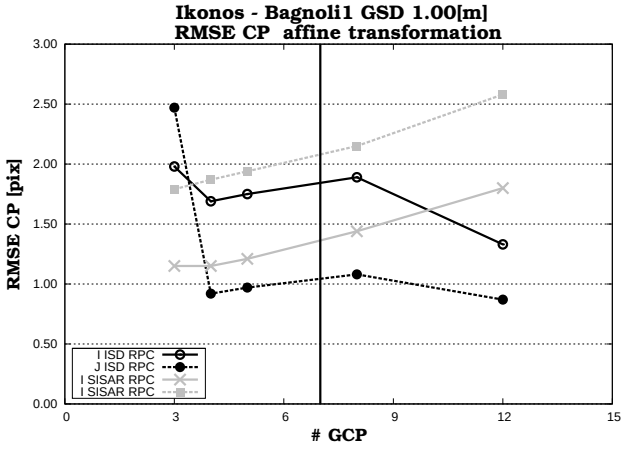


Figure B.62: Ikonos Bagnoli1 - Image accuracy obtained with an affine transformation for OrthoEngine software

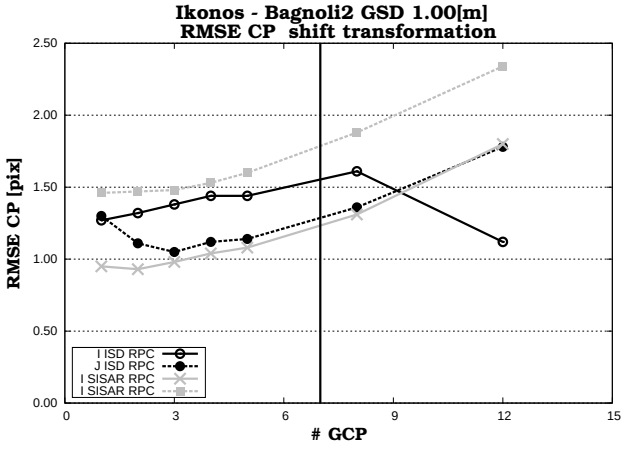


Figure B.63: Ikonos Bagnoli2 - Image accuracy obtained with a shift transformation for OrthoEngine software

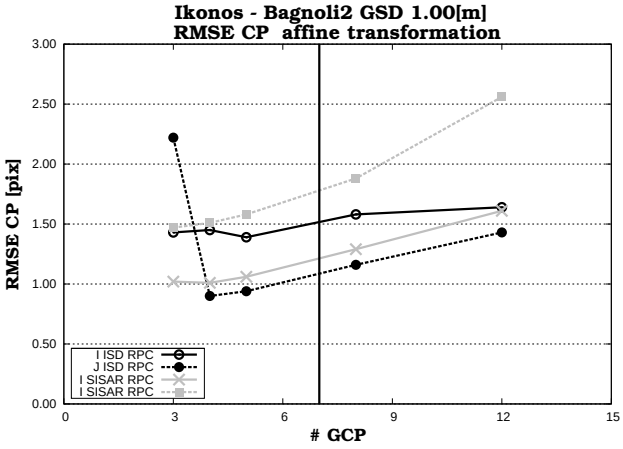


Figure B.64: Ikonos Bagnoli2 - Image accuracy obtained with an affine transformation for OrthoEngine software

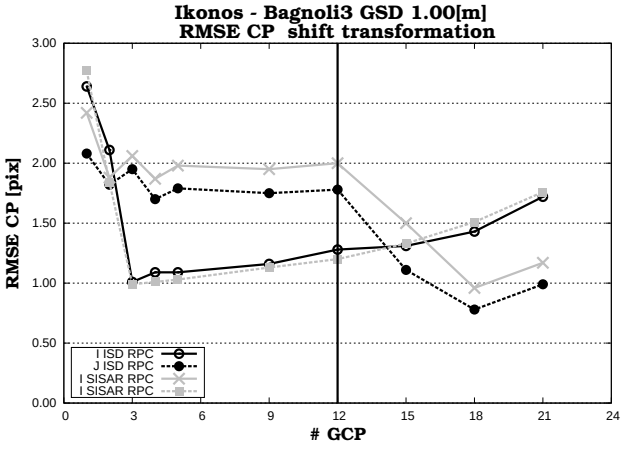


Figure B.65: Ikonos Bagnoli3 - Image accuracy obtained with a shift transformation for OrthoEngine software

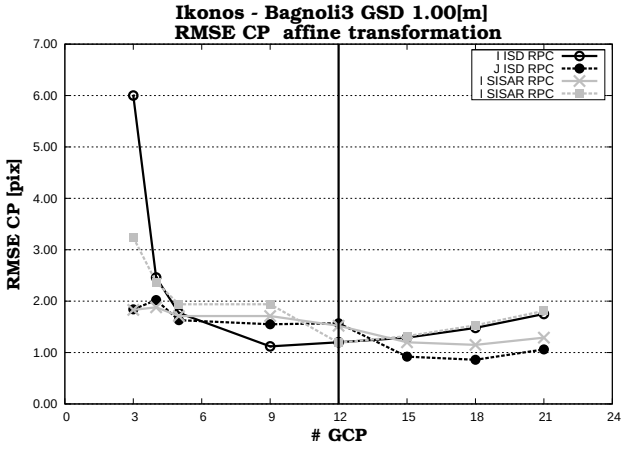


Figure B.66: Ikonos Bagnoli3 - Image accuracy obtained with an affine transformation for OrthoEngine software

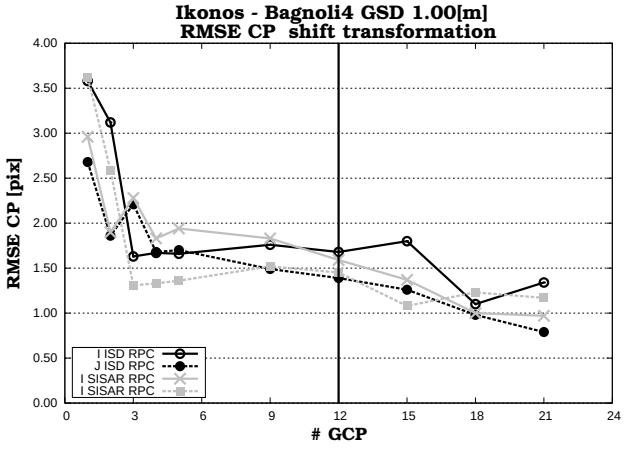


Figure B.67: Ikonos Bagnoli4 - Image accuracy obtained with a shift transformation for OrthoEngine software

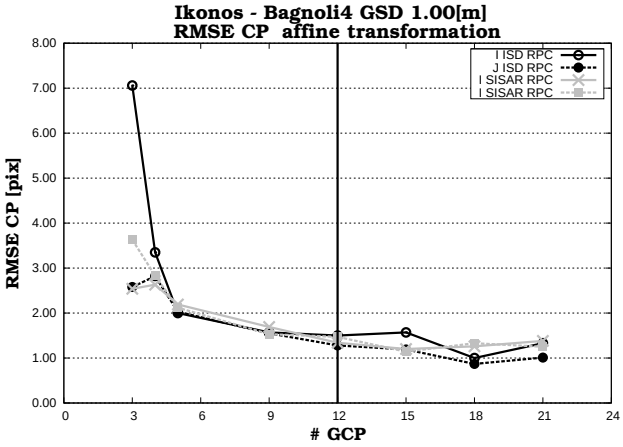


Figure B.68: Ikonos Bagnoli4 - Image accuracy obtained with an affine transformation for OrthoEngine software

B.4 Comparison between ISD RPCs and SISAR RPCs in Erdas software

B.4.1 QuikBird

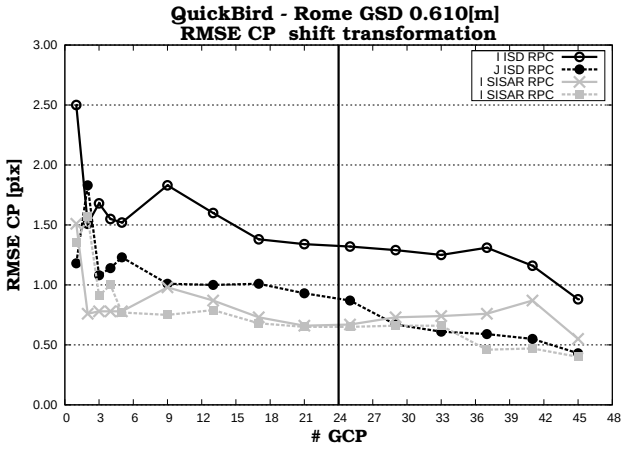


Figure B.69: QuickBird Rome - Image accuracy obtained with a shift transformation for Erdas software

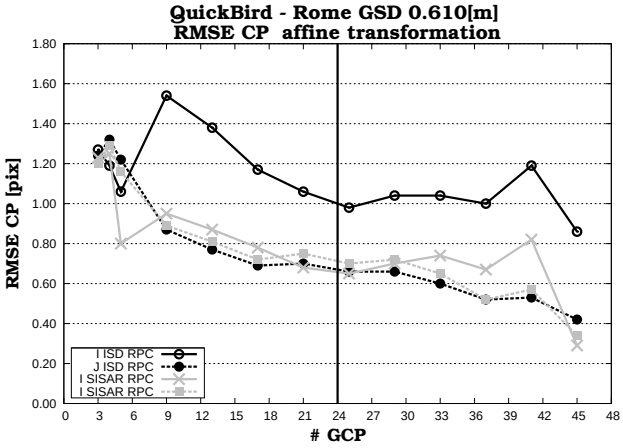


Figure B.70: QuickBird Rome - Image accuracy obtained with an affine transformation for Erdas software

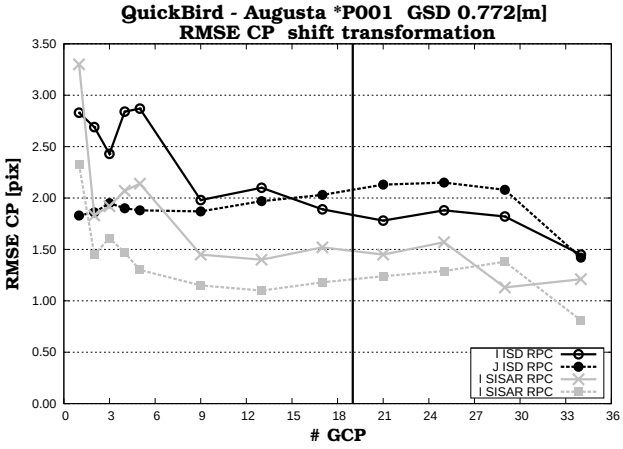


Figure B.71: QuickBird Augusta (*P001) - Image accuracy obtained with a shift transformation for Erdas software

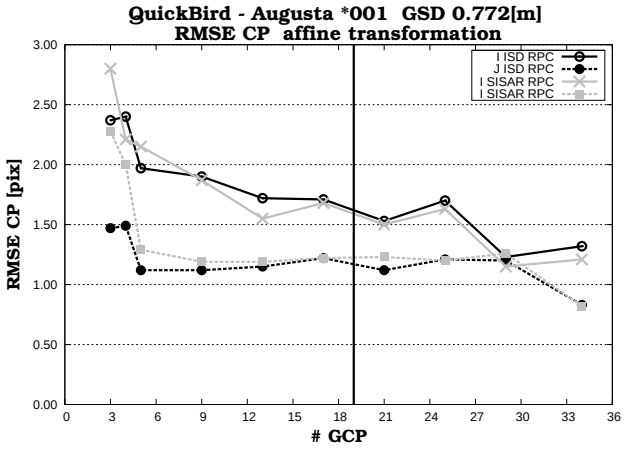


Figure B.72: QuickBird Augusta (*P001) - Image accuracy obtained with an affine transformation for Erdas software

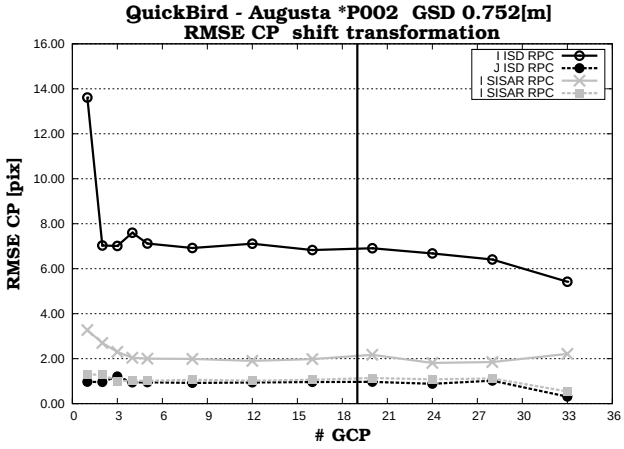


Figure B.73: QuickBird Augusta (*P002) - Image accuracy obtained with a shift transformation for Erdas software

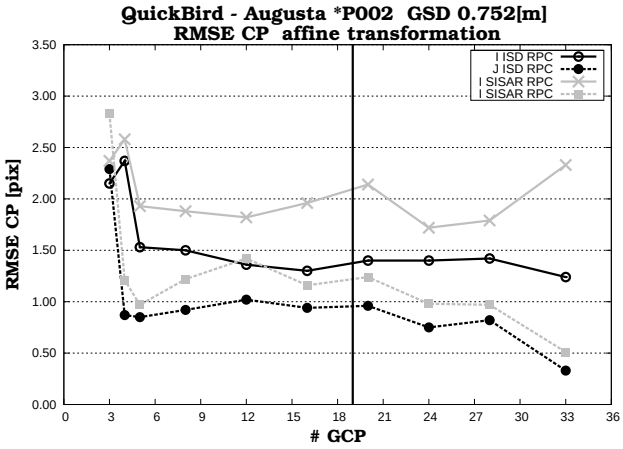


Figure B.74: QuickBird Augusta (*P002) - Image accuracy obtained with an affine transformation for Erdas software

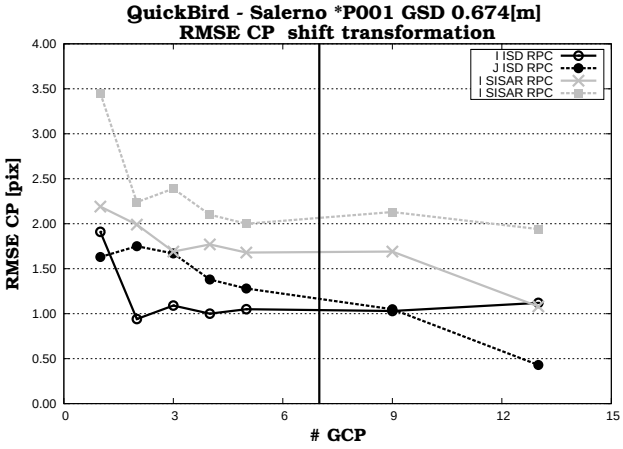


Figure B.75: QuickBird Salerno (*P001) - Image accuracy obtained with a shift transformation for Erdas software

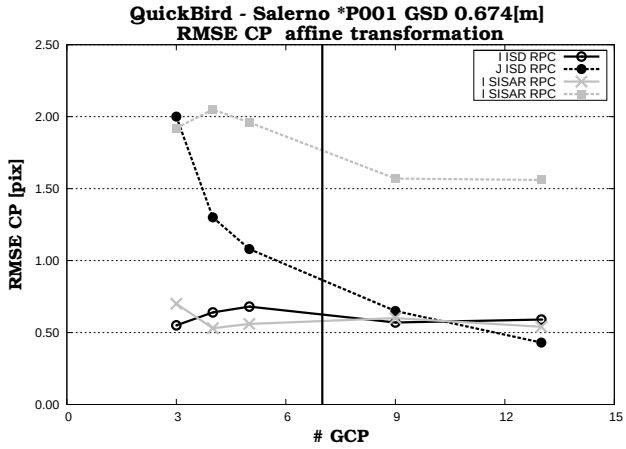


Figure B.76: QuickBird Salerno (*P001) - Image accuracy obtained with an affine transformation for Erdas software

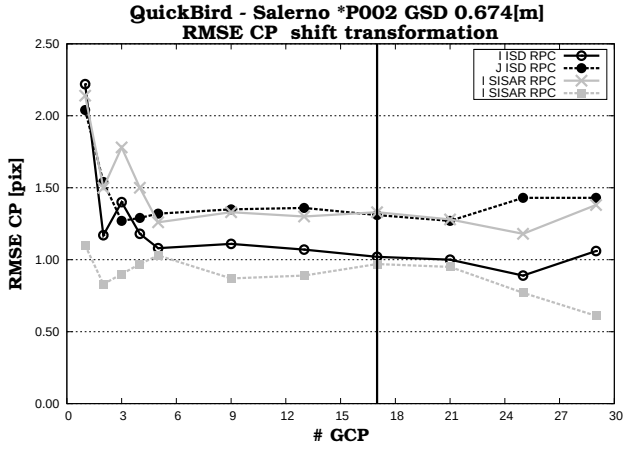


Figure B.77: QuickBird Salerno (*P002) - Image accuracy obtained with a shift transformation for Erdas software

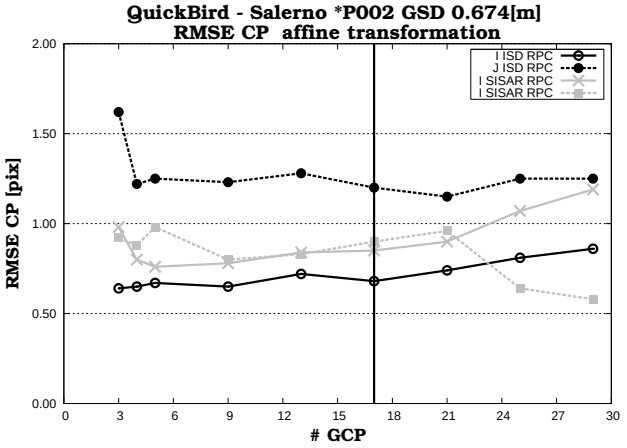


Figure B.78: QuickBird Salerno (*P002) - Image accuracy obtained with an affine transformation for Erdas software

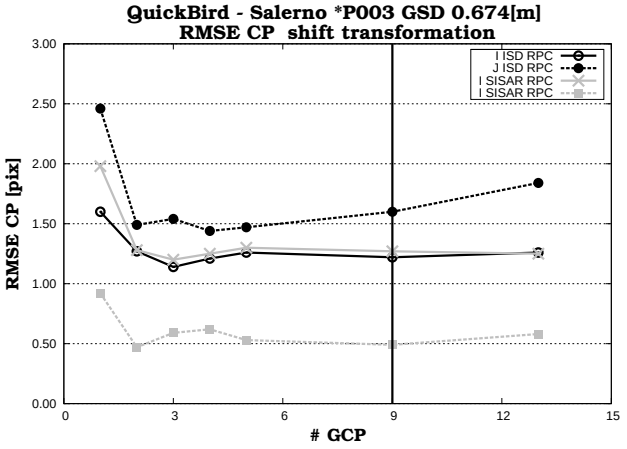


Figure B.79: QuickBird Salerno (*P003) - Image accuracy obtained with a shift transformation for Erdas software

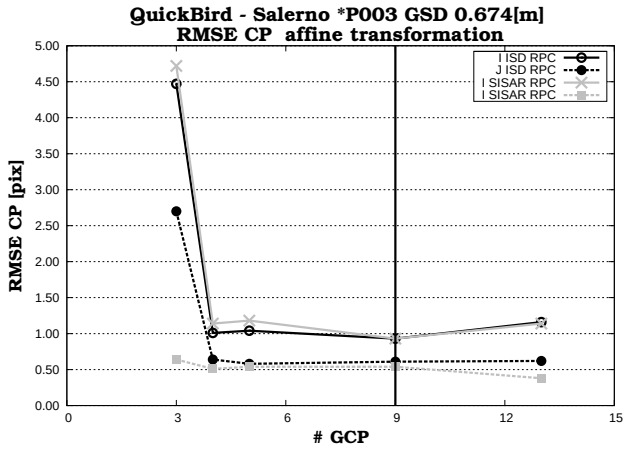


Figure B.80: QuickBird Salerno (*P003) - Image accuracy obtained with an affine transformation for Erdas software

B.4.2 Ikonos

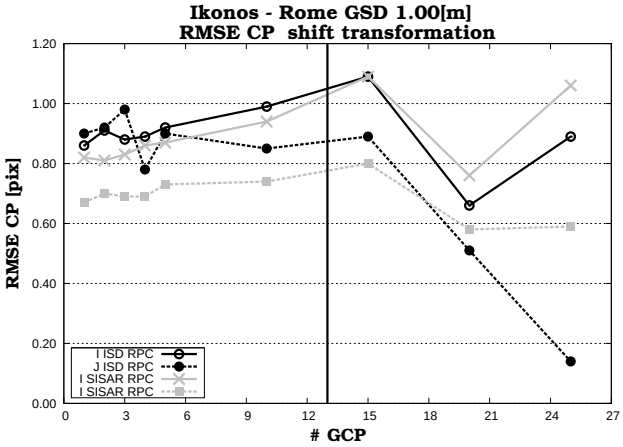


Figure B.81: Ikonos Rome - Image accuracy obtained with a shift transformation for Erdas software

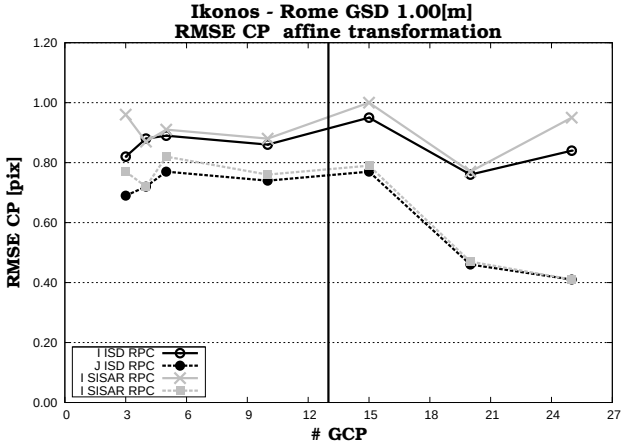


Figure B.82: Ikonos Rome - Image accuracy obtained with an affine transformation for Erdas software

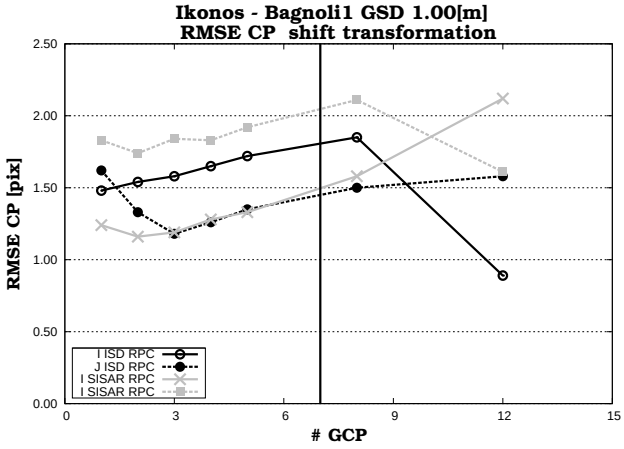


Figure B.83: Ikonos Bagnoli1 - Image accuracy obtained with a shift transformation for Erdas software

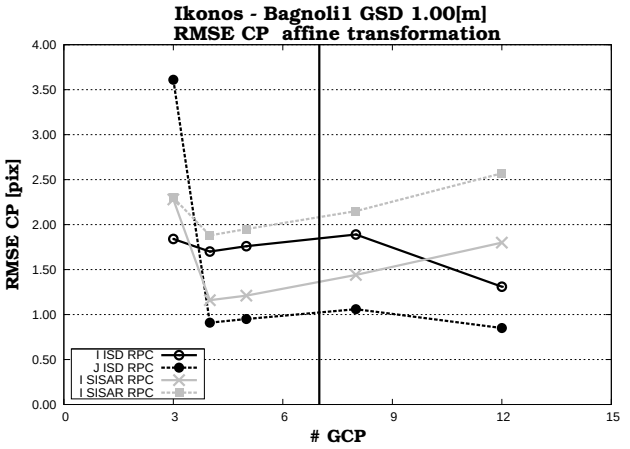


Figure B.84: Ikonos Bagnoli1 - Image accuracy obtained with an affine transformation for Erdas software

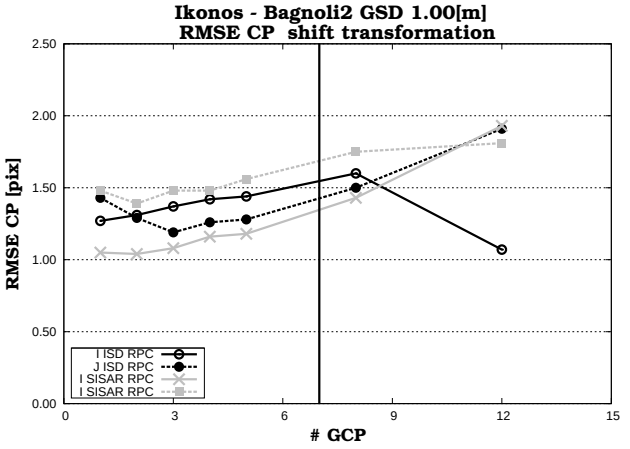


Figure B.85: Ikonos Bagnoli2 - Image accuracy obtained with a shift transformation for Erdas software

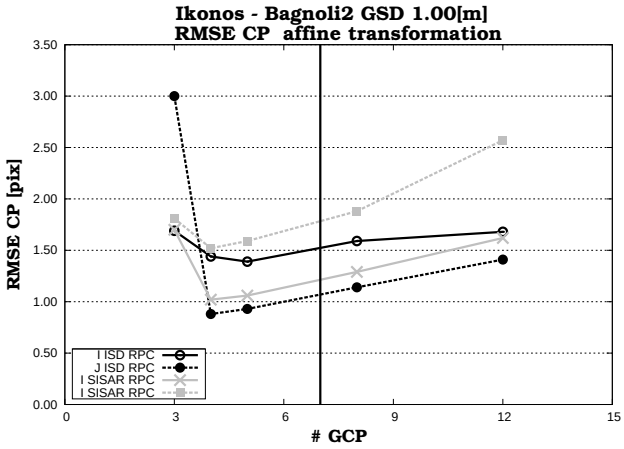


Figure B.86: Ikonos Bagnoli2 - Image accuracy obtained with an affine transformation for Erdas software

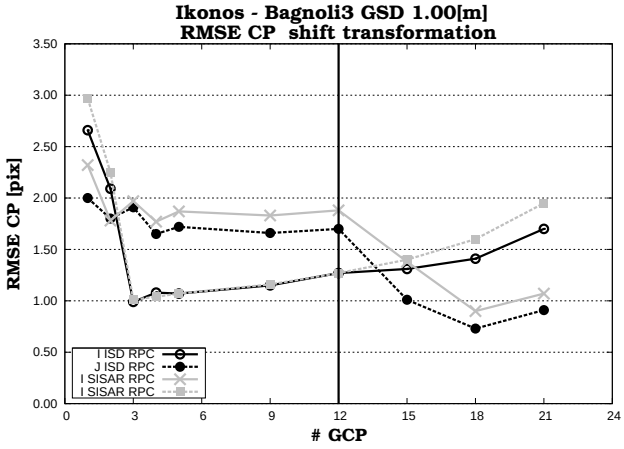


Figure B.87: Ikonos Bagnoli3 - Image accuracy obtained with a shift transformation for Erdas software

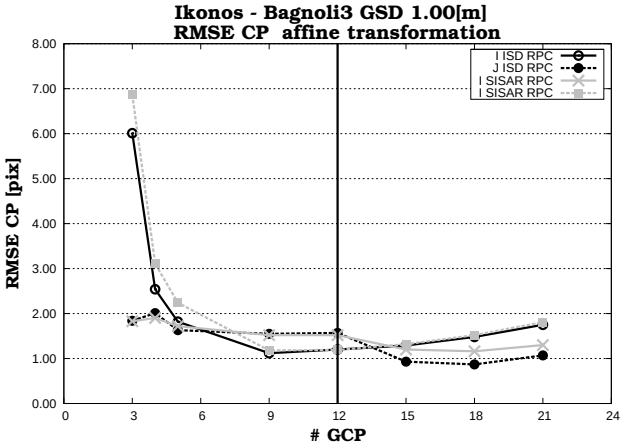


Figure B.88: Ikonos Bagnoli3 - Image accuracy obtained with an affine transformation for Erdas software

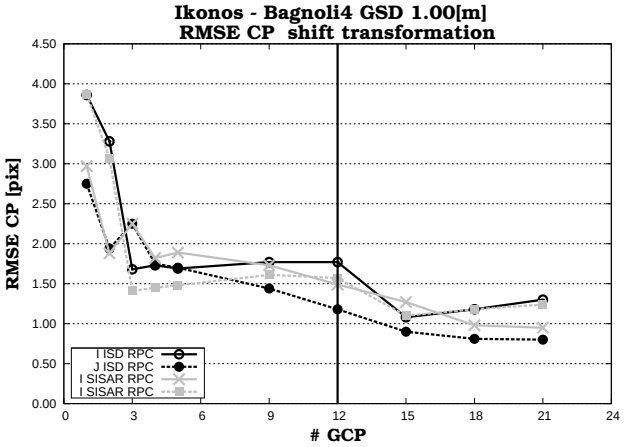


Figure B.89: Ikonos Bagnoli4 - Image accuracy obtained with a shift transformation for Erdas software

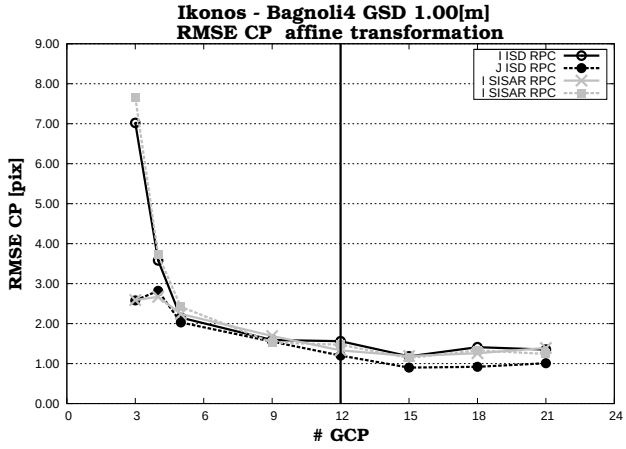


Figure B.90: Ikonos Bagnoli4 - Image accuracy obtained with an affine transformation for Erdas software

B.5 Comparison between RPCs model and Rigorous model

B.5.1 EROS A

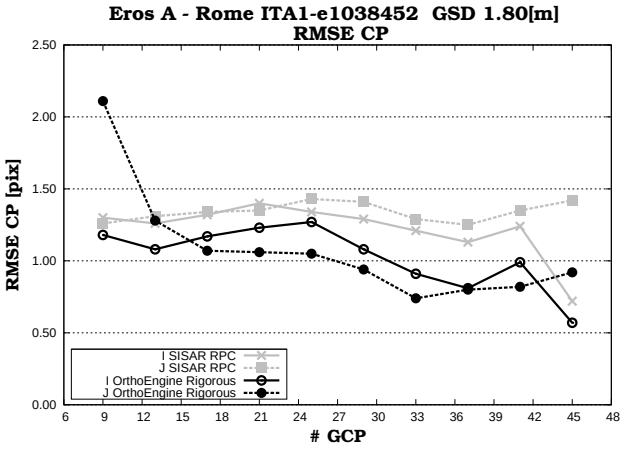


Figure B.91: EROS A R1 - Comparison between RPCs model and rigorous model

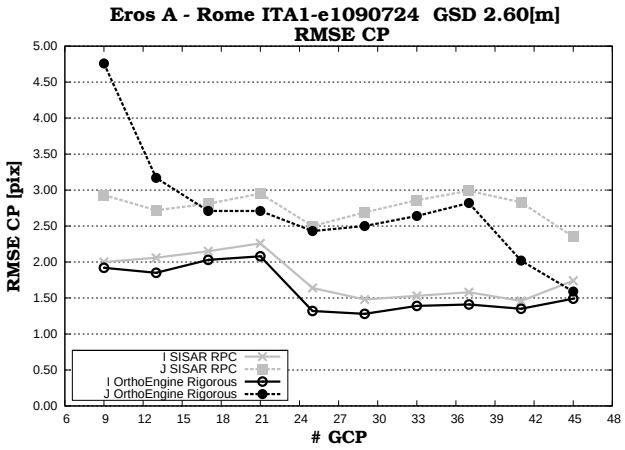


Figure B.92: EROS A R2 - Comparison between RPCs model and rigorous model

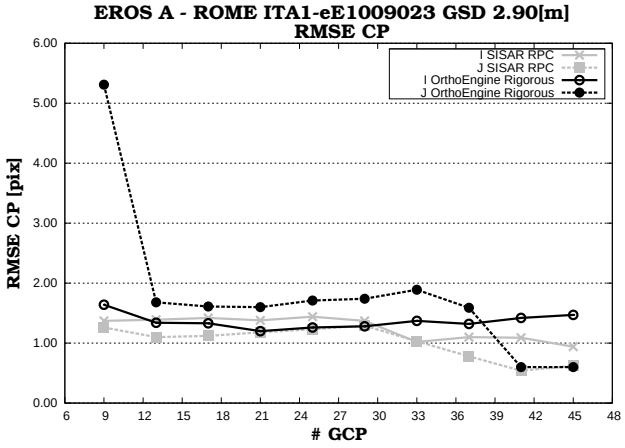


Figure B.93: EROS A R4 - Comparison between RPCs model and rigorous model

B.5.2 QuickBird

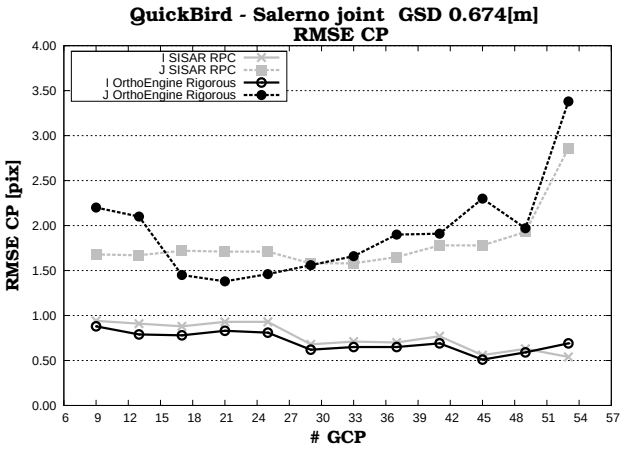


Figure B.94: QuickBird Salerno joint - Comparison between RPCs model and rigorous model

B.6 Stereo model via RPCs

B.6.1 Cartosat-1

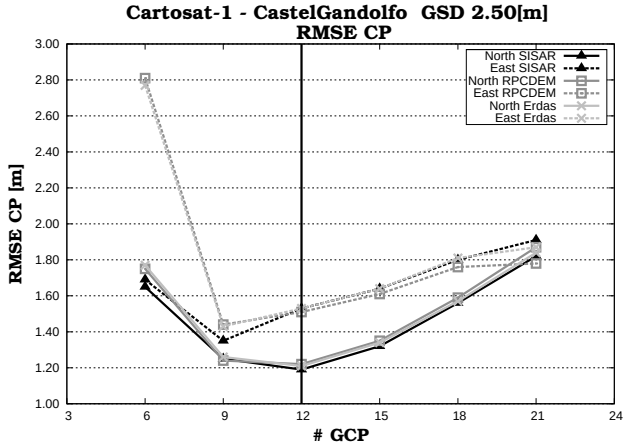


Figure B.95: Image accuracy vs. GCP number for Cartosat-1 stereopair of Castel-Gandolfo in North and East for all tested software

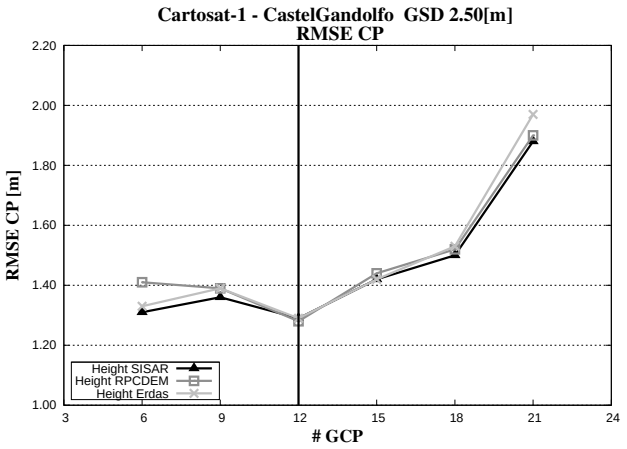


Figure B.96: Image accuracy vs. GCP number for Cartosat-1 stereopair of Castel-Gandolfo in Height for all tested software

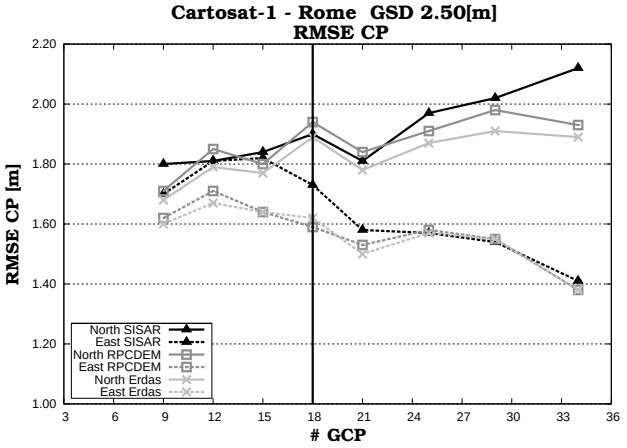


Figure B.97: Image accuracy vs. GCP number for Cartosat-1 stereopair of Rome in North and East for all tested software

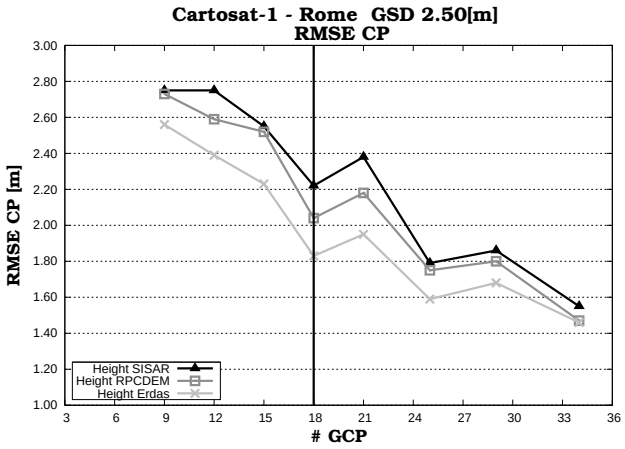


Figure B.98: Image accuracy vs. GCP number for Cartosat-1 stereopair of Rome in Height for all tested software

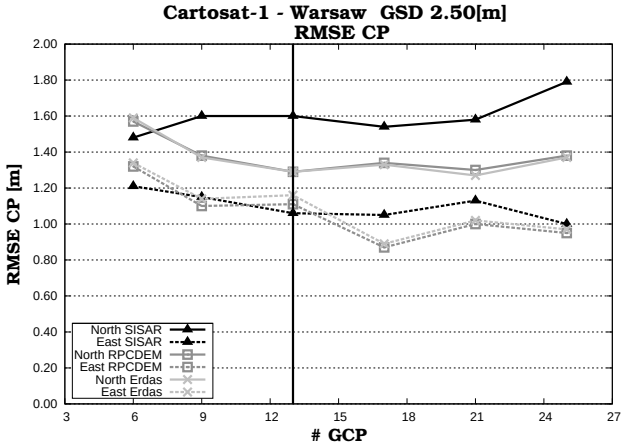


Figure B.99: Image accuracy vs. GCP number for Cartosat-1 stereopair of Warsaw in North and East for all tested software

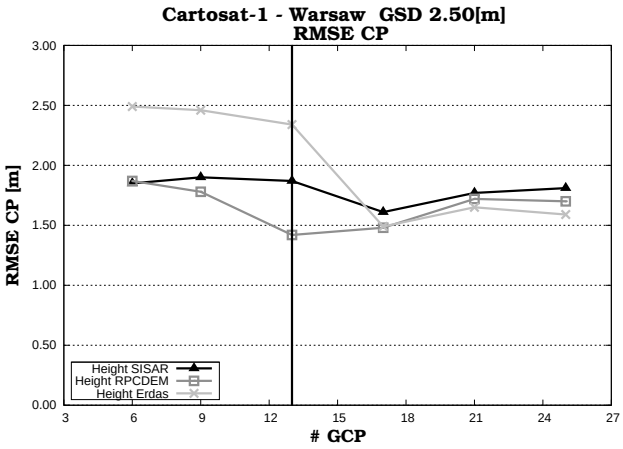


Figure B.100: Image accuracy vs. GCP number for Cartosat-1 stereopair of Warsaw in Height for all tested software

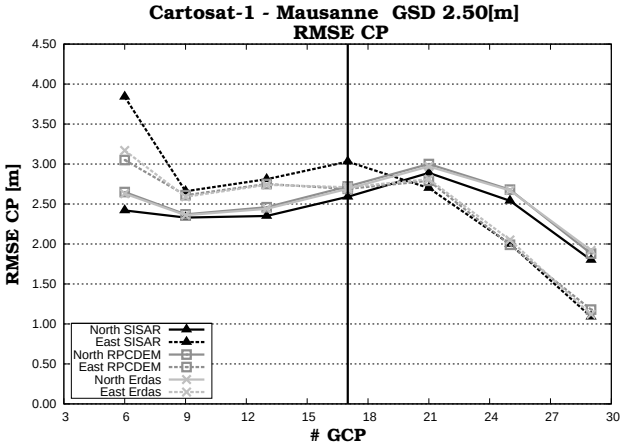


Figure B.101: Image accuracy vs. GCP number for Cartosat-1 stereopair of Mausanne in North and East for all tested software

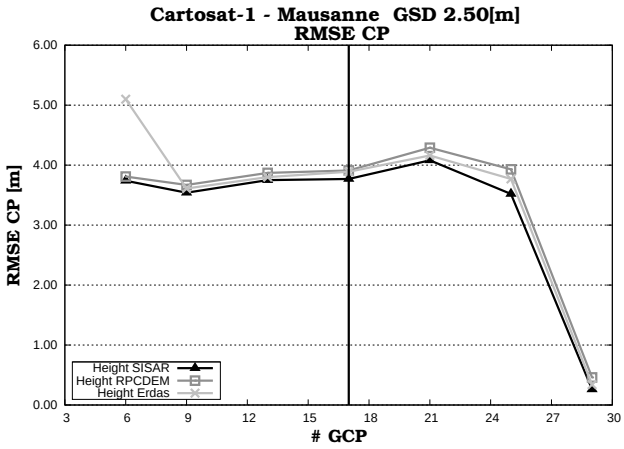


Figure B.102: Image accuracy vs. GCP number for Cartosat-1 stereopair of Mausanne in Height for all tested software

Appendix C

Acronyms

CP	Check Point
DEM	Digital Elevation Model
DITS	Dipartimento di Idraulica Trasporti e Strade
DLT	Direct Line Transformation
DOS	Department of Space
DSM	Digital Surface Model
ECEF	Earth Centered-Earth Fixed
ECI	Earth Centered Inertial
GCP	Ground Control Point
GPS	Global Position System
GSD	Ground Simple Distance
IERS	International Earth Rotation and Reference Systems Service
ISD	Image Support Data
LOS	Line of Sight
LS	Least Square
RMSE	Root Mean Square Error
RPC	Rational Polynomial Coefficient
RPF	Rational Polynomial Function
RTK	Real Time Kinematic
SISAR	Software per Immagini Satellitari ad Alta Risoluzione
SRTM	Shuttle Radar Topoghapy Mission
SVD	Singular Value Decomposition
TP	Tie Point

Acknowledgements

I would like to thank very much my Prof. Mattia Giovanni Crespi first of all for given me the opportunity to attend a PhD Course and for his encouragement and continuous support.
I also would like to thank Prof. Karsten Jacobsen for welcomed me in Leibniz University Hannover and for dedicated his precious time to help me.
Thanks to all the staff of the Geodesy and Geomatic Area of the University of Rome "La Sapienza" for their generous and varied contributions.

Bibliography

[1] D. Poli. Modelling of spaceborne linear array sensors. *Diss., Technische Wissenschaften ETH Zurich, Nr. 15894,IGP Mitteilung*, (85), 2005.

[2] P. D. Noerdlinger. Atmospheric refraction effects in earth remote sensing. *ISPRS Journal of Photogrammetry & Remote Sensing*, 54:360–373, 1999.

[3] W. M. Kaula. Theory of Satellite Geodesy. *Blaisedell Publishing Company*, 1966.

[4] T. Westin. Precision rectification of spot imagery. *Photogrammetric Engineering and Remote Sensing*, 56(2):247–253, 1990.

[5] A. Kleusberg P.J.G. Teunissen. Gps for geodesy. *Springer-Verlag. ISBN: 3-540-63661-7*, 1998.

[6] E. Wasle B. Hofmann Wellenhof, H. Lichtenegger. Gnss global navigation satellite system. *Spinger-Verlag. ISBN: 978-3-211-73012-6*, 2008.

[7] O. Montenbruck and E. Gill. Satellite orbits. *Springer, Berlin*, 2001.

[8] D.C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[9] H.A. Beyer. Geometric and radiometric analysis of a ccd-camera based photogrammetric close-range system. *PhD thesis, Institut fur Geodasie und Photogrammetry, Nr. 51, ETH, Zurich*, 1992.

[10] K. Jacobsen. Geometric calibration of space remote sensing cameras for efficient processing. *IAPRS*, 32:33–43, 1998.

[11] M. Crespi, F. Fratarcangeli, F. Giannone, K. Jacobsen, and F. Pieralice. Orientation of cartosat-1 stereo imagery. *Proceeding of EARSeL Joint Workshop Remote Sensing, New Challenges of High Resolution*, 2008.

[12] F. Giannone. A rigorous model for high resolution satellite imagery orientation. *PhD Thesis, Area di Geodesia e Geomatica Dipartimento di Idraulica Trasporti e Strade, Sapienza Università di Roma. Available online at: http://w3.uniroma1.it/geodgeom/geodgeomrw/downloads/tesi20dottorato/PhD%20Th%20Giannone.pdf*, 2006.

[13] M.A. Brovelli, M. Crespi, F. Fratarcangeli, F. Giannone, and E. Realini. Accuracy assessment of high resolution satellite imagery orientation by leave-one-out method. *ISPRS Journal of Photogrammetry & Remote Sensing*, 63(4):427–440, 2008.

[14] F. Sansò. Il trattamento statistico dei dati. *Editore CLUP, Milano*, 1989.

[15] F. Pieralice. Orthorectification of ikonos high resolution satellite imagery: definition, implementation and accuracy assessment of an original orientation model. *Degree thesis of the Sapienza University of Rome. Supervisors: M. Crespi. Not published*, 2007.

[16] M. Crespi, F. Fratarcangeli, F. Giannone, and F. Pieralice. Orientation of quickbird, ikonos and eros a stereopairs by an original rigorous model. *Proceeding of International Calibration and Orientation Workshop, EuroCOW 2008*, 2008.

[17] Y. Hu C.V. Tao. The rational function model. a tool for processing high resolution imagery. *Earth Observation Magazine*, 10(1):13–16, 2001.

[18] NIMA. The compendium of controlled extensions (ce) for the national imagery transmission format (version 2.1) nitfs technical board. 2000.

[19] Y. Hu C.V. Tao. 3d reconstruction methods based on the rational function model. *Photogrammetric Engineering & Remote Sensing*, 68(7):705–714, 2002.

[20] C. S. Fraser H.B. Hanley. Sensor orientation for high-resolution satellite imagery: further insights into bias-compensated rpc. *Available on line at: http://www.isprs.org/istanbul2004/comm1/papers/5.pdf*, 2004.

[21] Y. Hu C.V. Tao. A comprehensive study of the rational function model for photogrammetric processing. *Photogrammefric Engineering & Remote Sensing*, 67(12):1347–1357, 2001.

[22] Y. Hu C.V. Tao. Use of the rational function model for image rectification. *Canadian Journal of Remote Sensing*, 27(6):593–602, 2001.

[23] Y. Carbonneau T. Toutin, R. Chénier. 3d models for high resolution images: examples with quickbird, ikonos and eros. *Proceedings of ISPRS Commission IV Symposium, Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa*, pages 547–551, 2000.

[24] A. Neumaier. Solving ill-conditioned and singular linear systems: a tutorial on regularization. *SIAM Review*, 40(3):636–666, 1998.

[25] F. Fratarcangeli F. Giannone F. Pieralice M. Bianconi, M. Crespi. A new strategy for rational polynomial coefficients generation. *Proceeding EARSeL Joint Workshop Remote Sensing, New Challenges of High Resolution, Bochum (Germany)*, 2008.

[26] G. Strang and K. Borre. Linear algebra, geodesy and gps. *Wellesley-Cambridge Press, Wellesley*, 1997.

[27] G.H. Golub and C. F. VanLoan. Matrix computation. *The Johns Hopkins University Press, Baltimore and London*, 1993.

[28] W. H. Press, S. A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical recipes in c: The art of scientific computing. *Cambridge University Press (ISBN 0-521-43108-5). http://www.nr.com (accessed on April 10, 2006)*, 1992.

[29] P. J. G. Teunissen. Adjustment theory. series on mathematical geodesy and positioning. *Delft University Press*, 2001.

[30] Benciolini B. and Mussio L.

[31] M. Crespi, F. Giannone, and D. Poli. Analysis of rigorous orientation models for pushbroom sensors. applications with quickbird. *Proceedings of the ISPRS Commission I, WG V Meeting, 4 - 7 July, Paris,*, 2006.