ORTHOPHOTOGRAPHS

Javier A. Múgica, May 2008

How the orthophoto is built

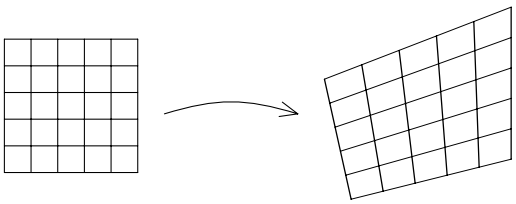
An orthophoto is an image that corresponds to a photograph taken at infinite distance above the Earth surface. This definition is valid for small areas, such as a small city. For greater areas the curvature of the Earth surface would be apparent —or at least measurable— and that is not what we understand as an orthophoto; that is, an orthophoto is *not* an orthogonal projection of a piece of the Earth surface.

Actually, the position of the points in an orthophoto is that corresponding to some cartographic projection. So an orthophoto has a projection, as maps do.

In order to start building an orthophoto two pieces of information have to be known: i) the orientation of the image and ii) the position in space of the points appearing in the image, i.e., the Digital Terrain Model (DTM).

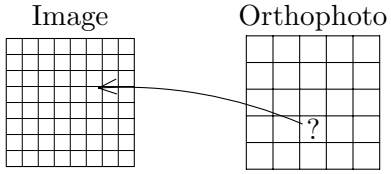
There are in principle two approaches for creating the orthophoto out of a given image. One of them consists in deforming the image so as to place each pixel in its position according to the projection:

fig. 1



The severe drawback of this solution is that we do not get a rectangular matrix of pixels, but instead a complicate pattern depending on the original tilt of the photograph and the formulas of the projection. Indeed, this pattern becomes completely irregular due to the relief of the ground. This makes the orthophoto very difficult to display on a screen and very difficult to handle by exploiting software as well as printing devices. Therefore the solution actually used is always the other one.

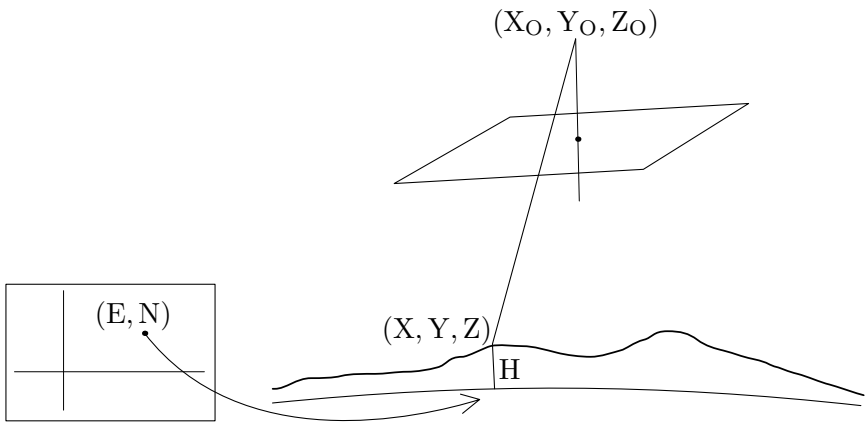
fig. 2



This second approach consists in finding the radiometric value that corresponds to the points of a square matrix in the projection; that is, of asking "what value do we have to assign to each point of the orthophoto?" (*fig.* 2).

This is solved point by point in the following way: Given the point with coordinates (E, N) of the orthophoto, whose radiometric value we want to find, we go back the projection to the space (the Earth) and therefrom to the image:

fig. 3



and pick the radiometric value of the point we have got to in the image.

It first has to be defined the size of the orthophoto pixels. It should be selected so as to match as closely as possible the image pixel.

Example: The mean photo height above the ground is 3 000 m, the focal length equals 152 mm and the pixel size is 10 μm. The ground pixel size that better fits the image pixel is then

$$0.01\,\frac{3000}{152} \approx 0.20\,\text{m}$$

The problem arises that we will not in general fall exactly at the center of a pixel when going from the point (E, N) to the corresponding point (x, y) in the image (we will later call (x', y') the points in the image, but for the time being let them be (x, y)). There are several ways of taking a value depending on the coordinates (x, y). this process is called resampling.

Resampling techniques

In order to get a correct understanding of the resampling process, the image pixels should not be thought of as points, but as squares. They are cells that tile the (x, y) plane of the image. The value of a pixel (for images this is a radiometric value) is thought when resampling as being achieved at the center of the pixel. To word it mathematically, the image is a function whose values are known at certain concrete points, the pixel centers, and elsewhere have to be guessed (interpolated) from the values at the known points.

Let the centers of the image pixels have integer coordinates, varying one unit per pixel. So a pixel is, for instance, the square $[149.5, 150.5] \times [223.5, 224.5]$, with center $(150, 224)$. For a given point (E, N) the corresponding (x, y) image coordinates would have been found and these are not in general integer values. The problem is to assign a value to the point (x, y) as a function of the surrounding pixel centers.

The easiest solution is to get the value of the nearest pixel center. This method is called **nearest neighbor**.

Example: Let the following matrix be a subset of an image.

	220	221	222	223
1034	83	84	88	90
1033	83	83	85	88
1032	82	83	84	86
1031	81	82	83	84

and let us suppose that we have found the point $(221.3, 1032.6)$ to be the one corresponding to $(\mathrm{E}, \mathrm{N}) = (100\,025, 80\,650)$. The nearest pixel center is $(221, 1033)$, so we assign its value —i.e., 83— to the point $(221.3, 1032.6)$ and hence to the point $(100\,025, 80\,650)$ of the orthophoto.

Note that the usual growing sense of the rows has been reversed. This is done so that the (x, y) coordinate system is oriented as the (E, N) coordinate system of the orthophoto.

The next method, ascending in complexity, is to perform a linear interpolation inside the square where the point (x, y) lies whose coordinates are pixel centers:

fig. 4

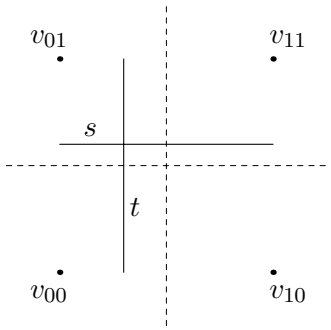
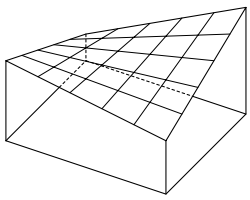


fig. 5



In *fig.* 4 the four dots are pixel centers; the cross point of the horizontal and vertical continuous lines is the (x, y) point which value we want, and the dashed lines are pixel boundaries. Let v_{00}, v_{01}, v_{10} and v_{11} be the values of the image at the pixel centers, as shown in the figure, and let (s, t) be the fractional parts of the coordinates (x, y).

The linear interpolation can be visualized as in *fig.* 5. Four vertical segments have been drawn at the pixel centers (the corners of the square) equal in length to the values of the image at those points. The top points of the segments have been joined in the same order as their bottom points are, forming a spatial quadrilateral. Opposite sides

in this quadrilateral are divided into equal ratios and the corresponding points joined. The result is a ruled, non developable surface —a hyperbolic paraboloid.

The numerical calculation is no more than a translation into algebra of this geometric construction. We first interpolate within the horizontal sides of the square:

$$v_{x0} = (1-s)v_{00} + sv_{10},$$
$$v_{x1} = (1-s)v_{01} + sv_{11},$$

and next within the vertical line:

$$v_{xy} = (1-t)v_{x0} + tv_{x1}.$$

Carrying out the operations we get to

$$v_{xy} = (1-s)(1-t)v_{00} + s(1-t)v_{10} + (1-s)tv_{01} + stv_{11}.$$

This is an average of the four values where each value's weight is the product of the two opposite segments to it in *fig.* 4.

This method is called **bilinear interpolation**.

Example: With the same numbers as in the previous example, compute the interpolated value by bilinear interpolation.

The values are

$$(x, y) = (221.3, 1032.6) \quad \Rightarrow \quad (s, t) = (0.3, 0.6),$$

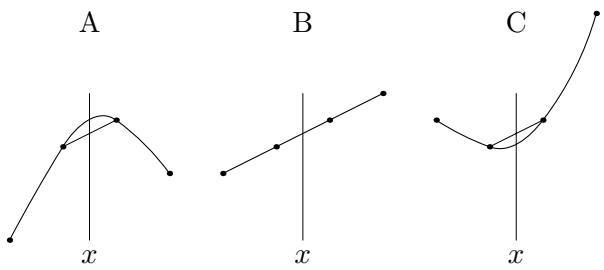
$$\begin{array}{ll} v_{01} = 83 & v_{11} = 85 \\ v_{00} = 83 & v_{10} = 84 \end{array}$$

The solution is

$$\begin{aligned} v_{xy} &= 0.7\,0.4\,v_{00} + 0.3\,0.4\,v_{10} + 0.7\,0.6\,v_{01} + 0.3\,0.6\,v_{11} \\ &= \ 0.28\cdot 83 \ + \ 0.12\cdot 84 \ + \ 0.42\cdot 83 \ + \ 0.18\cdot 85 \ = 83.48 \end{aligned}$$

This interpolation is of course better than the nearest neighbor (which does not interpolate at all), but it fuzzes the edges as a consequence of the averaging. In order to avoid this effect another method is devised which takes into account more pixels. Consider for instance the next three examples.

fig. 6

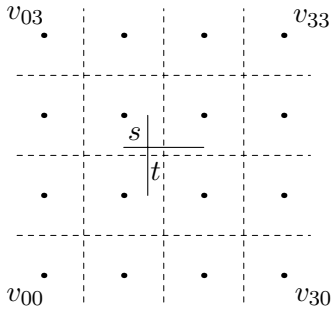


The dots represent again pixel centers. In order to simplify the example only one coordinate was considered. Let us suppose that the four pixels are at positions $x = 0$, 1, 2 and 3, and we want to interpolate the value at $x = 1.5$. In the three examples the pixels 1 and 2 have the same values, and therefore the linear interpolation would yield the same result. However, if we consider the four points and draw a smooth curve through them we see than in case A the middle third is higher than in case B and this in turn is higher than in C.

This example illustrates the fact that the higher the outer pixels the lower the interpolated value. So if the this value is to be computed as a weighted mean (and it will be), outer pixels should have negative weight.

This is accomplished by the **bicubic convolution**. There are other alternatives, but under a certain theoretical criterion this one is the best possible one.

fig. 7



There are a total of sixteen points involved in the interpolation (*fig.* 7). Let their values be denoted through v_{00} for the one at the lower left corner to v_{33} at the top right, and let s and t be as before.

The value according to this method is

$$v_{xy} = \sum p_u q_v \, v_{uv}.$$

That is, a weighted average where the weight of each pixel is obtained as the product of two factors, one corresponding to the column and other to the row. This was also the case for the bilinear interpolation.

The components p_u of the weights are functions of s:

$$\begin{aligned} p_0 &= -as(1-s)^2, & p_1 &= 1-(3-a)s^2+(2-a)s^3 \\ p_3 &= -a(1-s)s^2, & p_2 &= 1-(3-a)(1-s)^2+(2-a)(1-s)^3. \end{aligned}$$

And analogous formulas hold for q_v as functions of t.

The quantity a is a parameter of the method. In contrast to the preceding ones, the cubic convolution is not a single interpolation but a uniparametric family of interpolations, depending on one parameter. The choice $a = 0.5$ is usually a good one. For this value we have

$$\begin{aligned} p_0 &= -\tfrac{1}{2}s(1-s)^2, & p_1 &= 1-2.5s^2+1.5s^3 \\ p_3 &= -\tfrac{1}{2}(1-s)s^2, & p_2 &= 1-2.5(1-s)^2+1.5(1-s)^3. \end{aligned}$$

For example, v_{31} has the weight

$$p_3 q_1 = -\tfrac{1}{2}(1-s)s^2\,(1 - 2.5t^2 + 1.5t^3).$$

The whole summation can be written in matrix form:

$$\begin{pmatrix} q_3 & q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} v_{03} & v_{13} & v_{23} & v_{33} \\ v_{02} & v_{12} & v_{22} & v_{32} \\ v_{02} & v_{10} & v_{21} & v_{31} \\ v_{00} & v_{10} & v_{20} & v_{30} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

Example: Continuing with the same data, compute the interpolated value by bicubic convolution.

The values of p_u and q_v are

$$\begin{aligned} p_0 &= -\tfrac{1}{2}0.3\,0.7^2 = -0.0735 & q_0 &= -\tfrac{1}{2}0.6\,0.4^2 = -0.048 \\ p_1 &= 1 - 2.5\,0.3^2 + 1.5\,0.3^3 = 0.8155 & q_1 &= 1 - 2.5\,0.6^2 + 1.5\,0.6^3 = 0.424 \\ p_2 &= 1 - 2.5\,0.7^2 + 1.5\,0.7^3 = 0.2895 & q_2 &= 1 - 2.5\,0.4^2 + 1.5\,0.4^3 = 0.696 \\ p_3 &= -\tfrac{1}{2}0.7\,0.3^2 = -0.0315 & q_3 &= -\tfrac{1}{2}0.4\,0.6^2 = -0.072 \end{aligned}$$

We can check that $\sum p_u = 1$ and $\sum q_v = 1$ is satisfied.

There "only" remains to carry out the products and sums:

$$\begin{aligned} v_{xy} = \sum p_u q_v\, v_{uv} &= (-0.0735)(-0.048)81 + 0.8155(-0.048)82 + \cdots \\ &= 0.00353 \cdot 81 - 0.03914 \cdot 82 - 0.01390 \cdot 83 + 0.00151 \cdot 84 + \cdots + 0.00227 \cdot 90 \\ &= 83.29 \end{aligned}$$

The nearest neighbor technique, despite being very coarse, has the property of retaining the original values, which may be necessary for some applications, although it is unlikely.

Coordinate transformation

From (E,N,H) to (X,Y,Z)

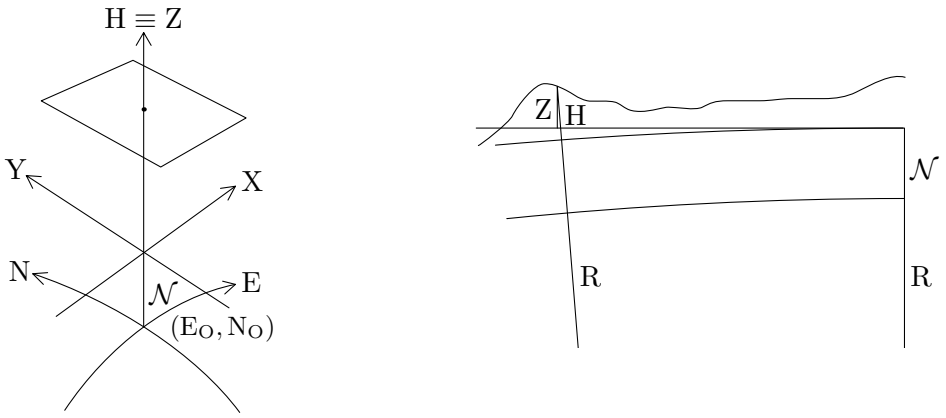
In order to find the value at a point (E, N) of the orthophoto we first get the H of that point in the DTM, and then we have to follow the path

$$(\mathrm{E}, \mathrm{N}, \mathrm{H}) \longrightarrow (\mathrm{X}, \mathrm{Y}, \mathrm{Z}) \longrightarrow (x, y)$$

The transformation from (E, N, H) to (X, Y, Z) takes into account the curvature of the Earth surface and the deformation of the projection. For satellite images the H may some times be ignored.

Let $(\mathrm{E_O}, \mathrm{N_O})$ be the (E, N) coordinates of the image projection center (*fig.* 8), and let k be the scale factor of the projection at the point $(\mathrm{E_O}, \mathrm{N_O})$. We will place the

fig. 8



origin of the (X, Y, Z) system at the point $(E_O, N_O, 0)$. We are supposing a conformal projection. If it were not, a more complicated transformation would take place.

If a point have (E, N) coordinates, the actual distance along the (E, N) axes from that point to the point (E_O, N_O) is

$$\frac{\Delta E}{k} \quad \text{and} \quad \frac{\Delta N}{k},$$

where $\Delta E = E - E_O$ and $\Delta N = N - N_O$. The axes run along the surface of the ellipsoid.

The difference between distances along the Earth surface and inside the tangent plane is a third order component and therefore negligible (even for some kilometers). Let $h = H + \mathcal{N}$ —therfore, h is the ellipsodal height. It can be shown (see the paper "Systèmes de référence") that, neglecting third order components,

$$\Delta E' = \frac{E - E_0}{k}, \qquad \Delta N' = \frac{N - N_0}{k}, \qquad S^2 = \Delta E'^2 + \Delta N'^2,$$

$$X = \frac{R + h}{R}\,\Delta E', \qquad Y = \frac{R + h}{R}\,\Delta N', \qquad Z = H - \frac{S^2}{2R}\,.$$

The coordinates of the projection center in the (X, Y, Z) system are $(0, 0, Z_O)$, where $Z_O = H_O$.

Example: Let an orthophoto be expressed in the GRS80 ellipsoid with UTM projection. The coordinates of an orthophoto pixel are

$$(720\,550, 4\,703\,100);$$

the H for that point according to the DTM is 680.6; the value of $\mathcal{N}$ at that point is 50, and the projection center coordinates are

$$(724\,030.23, 4\,702\,668.03, 6\,320.44)$$

All the values are given in meters. Find (X, Y, Z) for the point.

The scale factor for the UTM projection at the projection center coordinates is found to be

$$k = 1.0002176.$$

We will use $\mathrm{R} = 6\,378\,000$ m (adequate for these latitudes). We successively find:

$$\Delta\mathrm{E} = -3480.23, \qquad \Delta\mathrm{N} = 431.97, \qquad h = 730.6$$

$$\frac{1}{k}\frac{\mathrm{R}+h}{\mathrm{R}} = .9997825 \cdot 1.0001146 = 0.9998970$$

$$\mathrm{X} = -3480.23 \cdot 0.9998979 = -3479.87, \qquad \mathrm{Y} = 431.97 \cdot 0.9998970 = 431.93,$$

$$\mathrm{Z} = 680.6 - \frac{12293000}{12756000} = 679.64$$

From (X,Y,Z) to (x',y')

The (x, y) coordinates are related to $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ by the colinearity equations. With the usual notation, and noting that $\mathrm{X_O} = 0$ and $\mathrm{Y_O} = 0$,

$$x = -\mathrm{f}\,\frac{m_{11}\mathrm{X} + m_{12}\mathrm{Y} - m_{13}(\mathrm{Z} - \mathrm{Z_O})}{m_{31}\mathrm{X} + m_{32}\mathrm{Y} - m_{33}(\mathrm{Z} - \mathrm{Z_O})},$$
$$y = -\mathrm{f}\,\frac{m_{21}\mathrm{X} + m_{22}\mathrm{Y} - m_{23}(\mathrm{Z} - \mathrm{Z_O})}{m_{31}\mathrm{X} + m_{32}\mathrm{Y} - m_{33}(\mathrm{Z} - \mathrm{Z_O})}.$$

The image may have a geometric distortion. One may think that a good solution is to correct the images from distortion, so we forget about distortion once and for all. However this implies a resampling of the image. As a general rule,

> *The least the images are modified the better. Geometric transformation are better handled by software than by actually reshaping the photographs or mechanical devices.*

This applies not only to orthophoto generation but to the whole process of photogrammetric production and, indeed, to anything that requires precise measurements, like theodolites. Therefore, photogrammetric software must always take into account the distortion of the image when getting to it, or going form it to elsewhere.

So in order to get to the image coordinates where the point $(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$ actually appears, the distortion needs to be added, as well as the principal point coordinates if they are not zero:

$$(x', y') = (x, y) + \mathcal{D}(x, y) + (x_p^c, y_p^c)$$

Orthophoto tile

The most common situation when generating orthophotos is that a large orthophoto is composed from a set of photographs forming a block.

In its first step the orthophoto is generated in the same way as in the previous case —a pixel by pixel generation is performed. But now every point appears in at least two photographs and possibly in more, so a decision as to be made on which photograph to fetch the value from, or either to make some kind of average.

The averaging is discouraged because the different image points computed for a (E, N, H) point may not correspond exactly to the same ground detail. It should, but residuals always remain, and such an averaging would lead to blur edges.

The averaging being discarded, it is required to define a criterion for the choice of the photograph. Among all the photographs where the point appears, that which is closer to the vertical of the point will approximate better the orthogonal view. This leads to the criterion of maximum verticality.

A similar criterion is to select the photograph whose projection center is closer, in (E, N) coordinates, to the point. This is very similar to the previous criterion, the borders varying just a one or two pixels (ore none at all), and has the advantage that the photograph corresponding to each (E, N) of the orthophoto is independent of the H coordinate.

The appropriateness of verticality is also reflected in the flight itself. Flights for orthophoto generation have greater overlap among adjacent photographs, in order that for every ground point there is some photograph where it appears near the center. A typical overlap is 80%.

Two kind of problems appear at the borders. One is geometric, the other radiometric. The first one are small shifts that may appear at the border, specially noticeable at roads, dams, etc. However, if the orientations of the photographs are computed properly, the effect should be very little. The other is the difference in luminosity or color of the two photographs.

Both problems can be solved by manually drawing the choice borderlines, making them coincident with natural breaklines of the ground. The radiometric problem, however, is only diminished, and it may still be very noticeable. It is for this that radiometric adjustments are required.

Radiometric adjustments

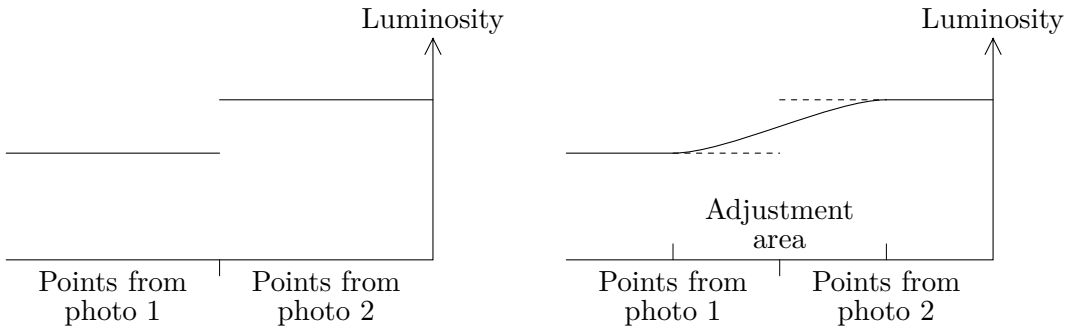
Two kind of radiometric adjustments are performed on the photographs. One is a constant per photo adjustment: darker photos are brightened and brighter ones are darkened; reddish ones are blued and bluish ones are red, etc.

In order for the computer program to detect the adjustments to be applied to each photograph it cannot just base its decisions on the average values of the photograph. There can be photographs that are darker just because they actually picture a darker area. For example, photos with large portion of water will be *much* darker than the others. Instead, the overlapping areas have to be compared.

This global adjustments may not be sufficient, specially if the camera objective has a systematic difference in luminosity from one area to another.

Fig. 9 pictures at the left a schematic graph of the luminosity along an edge perpendicular to a choice borderline. The picture in the right is brighter that that in

fig. 9



the left. A stream surrounding the borderline is selected and the radiometry of both images is modified so that they match at the choice border. the result is displayed at the right.

These adjustments can get more complex if the software implements better algorithms, but the idea remains the same.

The radiometric adjustment opens the possibility of integrating photos from different flights in the generation of a single orthophoto. This is specially useful for vast areas, where it may be very difficult that all the area be uncloudy simultaneously, or for high latitudes and ragged terrain, where the sun is a very short time high enough so as not to cast shadows.

Automatic generation of the DTM and the orthophoto

Once the orientation of the photographs is known, the DTM has been generated, the pixel size of the orthophoto defined and, if the user wants, manual choice borderlines have been drawn, the automatic generation of the orthophoto can proceed. The most complex parts are the organizing of the photographs, so as to create the (E, N) tile defining where each orthophoto pixel will be taken from, and the radiometric adjustments. The result is usually very good.

More difficult is the automatic generation of the DTM. It is carried out by correlation among images, making use of the parallax principle. But many difficulties arise in the process.

- *Huge amount of data.* This implies two problems. The volume of data itself and long processing times. The former is no longer a problem, given the ever more massive and cheaper storing devices. But the later is still of importance, and it is likely to continue being like that for a long time.

It is solved by and efficient data management (by the program that computes the DTM) and by good algorithms for correlation. For instance, the parallax of a given point is usually similar to that of the preceding one.

It is also important to choose the adequate grid size. Not all points can be correlated, and a too high density would increase very much the size of data and processing

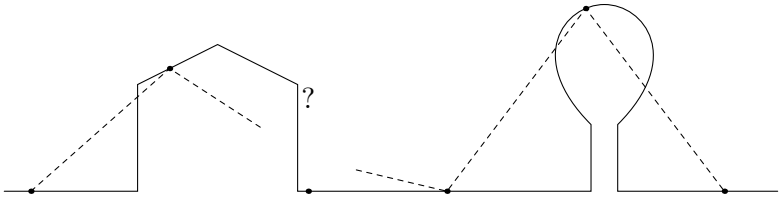
times while not improving the model. Indeed, it may worsen it, since small irregularities will appear more often.

- *Break lines.* The earth surface has lines along which the slope of the ground undergoes a clear change. The most prominent of these are rocky edges, rivers and wadis. While the software may try to detect them, it is extremely difficult and, anyway, it would need careful checking by the user, so in practice little is gained with respect to drawing them.

It is very important a conscious drawing of break lines, for a big part of the difference between a low-quality model and a good one is a correct drawing of brake lines.

- *Trees and buildings.* This is the crux of automatic DTM generation.

fig. 10



As *fig.* 10 shows, these accidents cause wrong points in the DTM. Thus, the model has to be manually edited after its generation and the process is not completely automatic.

Figure 10 exhibits two typical effects. One is the obvious prominences. The other is that vertical surfaces make some points hidden in some photographs, and the correlator will fail to correlate or will reach an incorrect result.

Some programs analyze the radiometry of the surroundings of the point to be correlated in order to detect trees or patterns similar to others selected by the user beforehand (training of the correlator), and if necessary displacing the point of correlation. This works for isolated elements, but not in a forest or in a city.

Let's start with the forest. With visible light it is impossible to see beyond the top of the trees, so it is these tops that have to be correlated. Afterwards (or before), the user may draw the limit of the forest, and by comparison with the adjacent ground the height of the trees is guessed, or it may be supplied by the user, and the whole area of the forest is lowered that height. This is more often than not the best that can be done.

With respect to the buildings, in a town or a city, it has first to be decided what we want: the actual frontier of the "solid" earth and the atmosphere, including the buildings and possibly the trees, or the street ground level. In the later case, the city may be delimited as was done for the forest, and the correlator may try to detect high and low points, and use only the low ones for the model.

In the other case, a good solution can be achieved diminishing the grid side within the city. This requires software capable of handling variable grid-size DTMs. If the city occupies a significant portion of the image, the grid size need be small in a large area,

and the data increases. There is however no other solution if we want the buildings to be represented properly.

Apart from other protruding buildings, such as chimneys, electric towers and others, there is the problem of bridges. Again, we have to decide whether we want the lower level or the higher one.

All these elements may raise as well the opposite problem. This happens when we want every element to appear. For example, when creating DTMs for aircraft navigation. In this case we even want the electric wires to appear, which will surely not be detected properly by the program. Indeed, wires is precisely what we would want most not to be missing. Consider for instance those crossing catenaries that fly more than one hundred meters above the bottom of the valley.

- *Moving elements.* When generating the DTM, the heights are computed based on parallax measurements. Moving elements change their position from one photograph to the next, hence appearing to have a greater or smaller parallax than the one corresponding to their height. If they happen to move along the flight direction, they may be correlated and generate a raised or sunk point.

In case they do not move along the flight direction, or even if they do, they may interfere the correlation of other points. In particular, a car top may be correlated with a flat building top or other similar features.

Also the trees leaves and branches move and change shape because of the wind. While not being significant for isolated trees —it does not solve the problem of trees, of course, but it does not worsen it either—, it can make the correlation inside forests very difficult, specially if the trees are dark. Given that in a forest the correlation is not precise anyway, a good solution is to enlarge the set of pixels being correlated —i.e., to use a 25×5 line instead of a 5×3 one, for instance, and/or not to use single pixels but groups of four pixels as if they were single pixels (this can be easily accomplished if the images are stored in a pyramidal structure).

- *Integrating the restitution and cartography in the DTM.* We have to distinguish here data coming from the same block of photographs used for the generation of the DTM and data from previous works or existing maps. In both cases the data is essentially a source of break lines, and it can improve the details of the DTM and at the same time saving time in the step of control and manual editing.

There is a very particular kind of data that deserves special treatment. It is contour lines. Most contour points are just points with known height, at no special location. However, the shape of the contours may be used to infer break lines. Many small watercourses, most of them usually without water, can be detected by analyzing the angles at a contour and with respect to the adjacent contours.

It is also possible to compare the contours arising from the DTM with the existing ones. They will not coincide because contours are not systematically used as break lines. Indeed, if we did so then there would be no need to correlate at all, and contours themselves, together with other break lines, would define the model. When we are providing contour lines to the DTM generation program they belong to other works or existing cartography, since we are generating the DTM precisely because we don't have it yet.

The integration of data coming from sources other than our block is more difficult because the coordinates of features in the source and in our work may not match exactly. The better the quality of both works the better the matching. This is the source of many problems for the generating software, that should analyze the supplied data and try to make it correspond with ours. There should be special care about differences in height, lest we get sunk roads or raised lakes or other incorrect results.

Integration of the orthophoto in the DTM

Once the orthophoto is generated in digital format, it may be printed or it may be further exploited. One possibility is the integration with the DTM. This can be used to create artistic perspectives as well as for some useful applications. We may, for example, include projected elements that do not yet exist on the ground, such as roads, pits, ... and make use of it in analysis of landscape impact or visibility studies (*fig* 11).

This integration also offers the possibility of creating animated trips along or above the Earth surface.

The underlying principle is that we should no longer think about the products as isolated results: maps, orthophotos, ... Nowadays, the digital storing of all the information allows its reuse and combination whenever it is needed and in any fashion permitted by the software and the ability of the cartographer.

fig. 11