SEMI-AUTOMATIC BUILDING RECONSTRUCTION INTEGRATED IN STRICT BUNDLE BLOCK ADJUSTMENT

Franz ROTTENSTEINER
Vienna University of Technology, Austria
Institute of Photogrammetry and Remote Sensing
fr@ipf.tuwien.ac.at

KEY WORDS: Buildings, Semi-automation, Data structures, Urban objects, Model-based processing.

ABSTRACT

This paper describes a new approach for semi-automatic building reconstruction integrated in hybrid photogrammetric adjustment. The new method uses a unique way of modelling building primitives by boundary representations (B-rep), making use of a special way of formulating surfaces, thus getting rid of the overhead parameters usually connected with B-rep models. In order to reconstruct a building, the user can select an appropriate building primitive from a knowledge base, adjust the parameters of that primitive to the actual data by interactively measuring points in digital images and determine the final building parameters by automated matching tools. Hybrid adjustment plays an important role in all stages of that process. This paper describes the mathematical model of hybrid adjustment as far as necessary to understand the new technique, it explains how the building models are formulated and how both the determination of approximations and the automated tools work. For all stages of the process, an example from a test project will be given, and first results of that project will be presented.

1 INTRODUCTION

The great demand for 3D city models, 3D GIS and virtual reality models collides with the enormous costs of data acquisition for these purposes. Thus, automation of extraction of buildings from various data sources is a very desirable and challenging yet equally difficult task because of the great number of possible building forms, even in a rather homogeneous cultural region such as Central Europe. However, due to the complexity of the task, fully automatic building extraction is not yet operational. That is why semi-automatic systems are currently being developed (Gülch et al., 1998, Veldhuis, 1998, Rottensteiner, 1998) which offer a compromise between the great demand for automation in data acquisition and the fact that the problem has not yet been completely solved. Semi-automatic systems provide a knowledge base of building primitives typical for a certain cultural region, and they offer tools for the automation of precise determination of the building parameters as well as for efficient interactive adaptations. The work flow for the reconstruction of a certain building looks as follows (Müller, 1998):

1. Interactive selection of an appropriate building primitive from the knowledge base by the human operator.
2. Interactive determination of approximations for the building parameters.
3. Automatic fine measurement and adjustment of the building parameters to the image data.
4. Visual inspection of the matching results and interactive editing in case the automation tools failed.

One of the most important issues connected with building extraction is the internal representation of model knowledge. Model knowledge about buildings can be represented in several ways (Brunn, 1998). **Parameterised models** are widely used for simple building primitives such as hip roof or saddle back roof buildings. The topology of these primitives is provided by the knowledge base, only the geometrical parameters have to be adjusted. **Generic building models** on the other hand provide a construction rule rather than a fixed topology. The simplest form of a generic building model is a prismatic model: prismatic building is characterized by two horizontal planes (roof and floor) and a set of n vertical planes (walls). Parameterised models can only be provided for simple building types. More complex buildings have to be modeld by a combination of simple primitives. This means that tools for the combination of the basic primitives are required (Müller, 1998).

Another important topic is concerned with the internal representation of buildings. Buildings are characterized by regular shapes such as vertical and/or parallel planes, rectangles or symmetrical surface parts. Boundary representations (B-rep) and Constructive Solid Geometry (CSG) are widely used to represent buildings in the computer (Brunn, 1998).

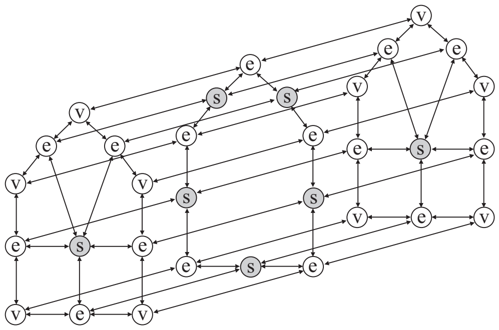

Figure 1: B-rep of a simple building model: a graph with nodes of type s (surfaces), e (edges) and v (vertices)

B-rep offer a very flexible tool for modelling man-made objects. They consist of surfaces, edges and points and the topological relations of these features. The surfaces, the edges and the vertices are the nodes of a labelled graph, and the direct neighbourhood relations are described by the edges of the graph (figure 1): Edges have two neighbouring vertices (starting point and end point) and two neighbouring surfaces which intersect at the edge. Surfaces thus have a set of neighbouring edges by which they are bordered. Vertices have a set of neighbouring edges which intersect at them.

CSG provides solid 3D primitives which are described by a set of parameters reflecting the object dimensions, e.g. width, length, height (Müller, 1998).

The CSG primitives are simple objects such as cubes, boxes or tetrahedrons, and more complex objects are composed of a set of primitives by logical operations: union, intersection and difference. It is the main benefit of CSG that a minimum set of parameters can be chosen for the description of each primitive. Symmetries are thus modelled implicitly, which can improve both the reliability and the speed of building reconstruction (Veldhuis, 1998). However, CSG is not as flexible as boundary representations: the applicability of CSG depends on the primitive set which can be used (Müller, 1998). If an inappropriate set of primitives is chosen, object modelling using these primitives will become difficult.

In this paper, we will describe a new system for semi-automatic building extraction from digital aerial images which is characterized by a unique way of modelling based on B-rep and the integration of a hybrid photogrammetric adjustment system. We use this method both to provide the knowledge base and to represent the building in the computer during geometrical reconstruction. We can avoid the over-parameterisation which is often attributed to B-rep due to an implicit formulation of symmetry assumptions so that in the end only a minimum set of parameters is required for each building primitive. Robust hybrid adjustment plays a key role in all stages in the work flow, and it is closely connected to the way we represent the buildings. That is why we describe the mathematical model on which adjustment is based in section 2. Using that background, our modelling technique will be described in section 3. After that, the work flow of building extraction will be explained, and an example will be worked out. Section 4 deals with the way the approximate building parameters can be determined interactively. In section 5 we will describe the algorithm for automatical fine measurement. First results from a test project will be given in section 6.

2 THE MATHEMATICAL MODEL OF HYBRID ADJUSTMENT

The hybrid photogrammetric adjustment system ORIENT has been developed at our institute since the mid-seventies. It offers the possibility of simultaneous hybrid least squares adjustment of various types of observations, among others image co-ordinates, control points, model co-ordinates and fictitious observations of points belonging to polynomial surfaces and/or 3D curves. In addition, ORIENT offers two blunder detection techniques: robust estimation and data snooping (Kager, 1989). ORIENT distinguishes two basic types of data, i.e. **observations** and **parameters**, the latter being the unknowns to be determined in adjustment. All data are stored in “rooms”. The term “room” can be literally understood to be a 3-D room with a 3-D Cartesian co-ordinate system attached to it. Each “observation room” (e.g. a photo) contains a list of points and references to parameters of a transformation between itself and a special “parameter room” called “Reference System” which is attached to the object co-ordinate system. The co-ordinates of the points in observation rooms are treated as observations in least squares adjustment. The transformation is described by a **mapping function**. Both number and interpretation of the parameters of that mapping function depend on the observation type.

2.1 The mapping functions

ORIENT treats all types of observations uniformly by using basically the same mapping function for all room types. The basic formula relating the observed point $\mathbf{p} = (u, v, w)^T$ to the object point $\mathbf{P} = (X, Y, Z)^T$ is given by the spatial similarity transformation (Kraus, 1997):

$$\mathbf{M} \cdot [\mathbf{p} - \mathbf{p}_0(adp)] = \lambda \cdot \mathbf{R}^T(\theta) \cdot (\mathbf{P} - \mathbf{P}_0) \qquad (1)$$

In equation 1, $\mathbf{p}_0 = (u_0, v_0, w_0)^T$ is called the interior reference point which may be modified by the additional parameters adp. $\mathbf{R}^T(\theta)$ is a transposed 3×3 rotational matrix which is computed from three rotational angles θ. θ can be parameterised in several ways, the most common one being the typical parameterisation of aerial photographs: $\theta = (\omega, \phi, \kappa)^T$. The scale factor between the observation and the object co-ordinate systems is λ. $\mathbf{P}_0 = (X_0, Y_0, Z_0)^T$

is called the exterior reference point, and finally, $\mathbf{M} = diag(m_u, m_v, m_v)$ is a mirror matrix containing the mirror coefficients $m_i = \pm 1, i \in \{u, v, w\}$ for the u, v and w axis, respectively.

With the exception of $\mathbf{M}$ which describes a property of the observation co-ordinate system, all groups of parameters in equation 1 can be determined in adjustment. Basically, these groups of parameters appear in the mapping functions of all observation types, but they might obtain different interpretations and/or be given constant default values. In the following sections, we want to describe the specific mapping functions for three types of observations which are of special interest in the context of semi-automatic building extraction.

2.1.1 Perspective observations (Photos). For observed photo points, the mapping function is a perspective transformation. The observation co-ordinate system is the camera co-ordinate system, and the third co-ordinate of the observed point $\mathbf{p}$ is 0. $\mathbf{P}_0$ is the projection centre in the object co-ordinate system, its camera co-ordinates being $\mathbf{p}_0 = (u_0, v_0, f)^T$ with f being the focal length. The additional parameters adp describe the distortion parameters. The influence of camera distortion is modelled by polynomials describing small variations of the principal point $(u_{pp}, v_{pp})^T$: $u_0 = u_{pp} + du_0(adp, u, v)$ and $v_0 = v_{pp} + dv_0(adp, u, v)$. The scale λ describes the location of $\mathbf{P}$ on the projection ray and thus is no longer equal for all points. By dividing the first two lines of equation 1 by the third one and by doing some re-arrangement, we finally obtain the well-known equations for central projection (Kraus, 1997).

2.1.2 Surface observations. Surfaces, too, are described in a local observation co-ordinate system. Again, the transformation formula is given by equation 1 with the additional assumption $\lambda = 1$. In this section, we will use the short-hand $\mathbf{p}_R = (u_R, v_R, w_R)^T = \mathbf{R}^T \cdot (\mathbf{P} - \mathbf{P}_0)$ for the right-hand side in equation 1. The observation "A point $\mathbf{P}$ is on a surface" can be expressed as "$\mathbf{P}$'s distance from that surface is observed to be 0". We do not use the Euclidean distance but its projection to one of the axes of the observation co-ordinate system. The interior reference point $\mathbf{p}_0$ receives a special interpretation: we consider its components to be polynomial functions of degree n of two components of $\mathbf{p}_R$, and these polynomials provide the equation of that surface in the observation co-ordinate system, e.g. $w_0 = w_0(adp, u_R, v_R)$. Similar considerations can be made for u and v, so that the observation that a point P is on a surface can be formulated in one of the following ways:

$$\begin{aligned} u &= 0 &= m_u \cdot u_R + \sum_{j,k=0}^{n,n} a_{jk} \cdot (m_v \cdot v_R)^j \cdot (m_w \cdot w_R)^k \\ v &= 0 &= m_v \cdot v_R + \sum_{i,k=0}^{n,n} b_{ik} \cdot (m_u \cdot u_R)^i \cdot (m_w \cdot w_R)^k \\ w &= 0 &= m_w \cdot w_R + \sum_{i,j=0}^{n,n} c_{ij} \cdot (m_u \cdot u_R)^i \cdot (m_v \cdot v_R)^j \end{aligned} \tag{2}$$

The additional parameters adp from equation 1 are replaced by the polynomial coefficients a_{jk}, b_{ik} and c_{ij} which describe the surface in the observation co-ordinate system (Kraus, 1997). An application is free to decide which of the coefficients are to be used for a certain surface: the list of additional parameters may consist of any subset of these coefficients, and the others are 0. It is one of the benefits of this way of mathematical formulation that geometrical constraints between surfaces can be modelled with it: By assigning identical surface coefficients and transformation parameters $(\mathbf{P}_0, \theta)$, but different values of $m_i, i \in \{u, v, w\}$, to different surfaces, symmetries with respect to the co-ordinate planes of the observation co-ordinate system can be modelled. By assigning identical coefficients and identical but unknown rotations θ to a set of surfaces, parallelism can be enforced, and rectangularity of two planes can be obtained by formulating them as being the $uv-$ and the vw-planes of the same co-ordinate system, respectively. 3D polynomial curves can be formulated as the intersection of two surfaces by using a set of two equations 2, and a 3D point is determined by the intersection of three surfaces, i.e. by all three equations 2. A point on a curve gives support to the parameter sets of both surfaces. As the attachment of parameter sets to observations is done by reference, one set of surface parameters can be used for more than one curve.

2.1.3 Observed parameters. The parameters of the mapping function 1 can usually not be observed, the exception being the object co-ordinates $\mathbf{P}$, which are considered to be observed for control points. However, if a parameter might not be determinable from other observations, an observation for that parameter might be useful to avoid singularities. With respect to observed parameters, equation 1 degenerates to $\mathbf{p} = \mathbf{P}$ assuming the parameter co-ordinate system (the object co-ordinate system in the case of control points) to be identical to the observation co-ordinate system. This equation can be applied to all groups of parameters, thus, there are observed rotation angles, observed surface coefficients, etc.

2.2 Hybrid adjustment

All observations are simultaneously adjusted in a hybrid least squares adjustment. The **functional model** of adjustment can be derived from the mapping functions explained in section 2.1: The expectation of $\mathbf{p}$, $\mathbf{E}(\mathbf{p})$, is a function $\mathbf{f}$ of the unknown parameter sets $\mathbf{p}_0$, adp, $\mathbf{P}_0$, θ, and $\mathbf{P}$:

$$\mathbf{E}(\mathbf{p}) = \mathbf{p} + \mathbf{r} = \mathbf{f}(\mathbf{p}_0, adp, \mathbf{P}_0, \theta, \mathbf{P}) \tag{3}$$

In equation 3, $\mathbf{r}$ is the vector of residuals. Note that equation 3 has between one and three components, depending on the observation type: an image point gives two equations ($\mathbf{p}$ has two components, i.e. the measured image co-ordinates), a surface observation gives one equation, and an observed parameter can give between one and three. The functions $\mathbf{f}$ can be derived easily from the mapping functions described in section 2.1. Equations 3 have to be linearized using approximate values for all parameters, which means that adjustment has to be performed iteratively. The **stochastical model** of adjustment is given by the weights $W_i = c^2/\sigma_i^2$, σ_i and c being the a priori r.m.s. errors of the observation i (an image co-ordinate or a point's distance from a surface or curve) and of the weight unit, respectively.

As observation rooms contain references to mapping parameters rather than the parameters themselves, it is easily possible to assign the same parameters to different rooms, thus yielding, e.g., two observation co-ordinate systems to be parallel or a set of photos to have the same inner orientation. In addition, ORIENT gives an application every freedom to decide which parameters are to be determined in a certain adjustment step and which are to be considered constants. However, there are dependencies e.g. between the linear coefficients of the surface observations (a_{jk}, b_{ik}, c_{ij}) and the rotations or between the constant coefficients and $\mathbf{P}_0$. With respect to curves, care has to be taken on the determinability of the coefficients of the intersecting surfaces. For example, thinking of a straight line being the intersection of two planes, the tilts of the planes orthogonal to the line either have to be determined by other observations or have to be declared constant. Another important property of our adjustment model is the fact that no homologous points are required to determine a curve in object space: a curve will be determined by intersection of two or more bundles of rays.

2.2.1 Robust estimation. ORIENT offers the possibility to detect gross errors in the data by means of robust estimation by re-weighting the observations depending on their residuals in the previous adjustment: As soon as convergence has been achieved using the original weights W_i, iteration will start again with the weight $W_{i,n+1}$ of observation i in adjustment $n+1$ being modulated depending on the size of the normalised residual $d_{i,n} = \frac{r_{i,n}}{\sigma_i}$ of that observation in iteration n (Kraus, 1997):

$$W_{i,n+1} = W_i \cdot \frac{1}{[1 + (\frac{d_{i,n}}{h})^4]^2} \tag{4}$$

Normalisation of residuals is necessary to be able to compare the residuals of different observation types. Parameter h is the size of a normalised residual causing weight modulation to give an observation half its original influence. If redundancy is great enough, blunders, i.e. observations not fitting to our mathematical model with the accuracy we expect, will successively lose influence by receiving a lower weight. However, an observation with a low influence according to equation 4 can be rehabilitated in the next iteration step if another one, this time the true blunder, has in the meantime been eliminated. Convergence speed depends on h; we usually select h a bit smaller than the greatest normalised residual and set $W_{i,n+1} = 0$ for $d_{i,n} > h$. Adjustment is repeated with h being reduced at each step until h reaches a given threshold, e.g. $h = 3$. In order to make this method of robust estimation work, a high redundancy is required, and the number of outliers should not exceed 30 %.

3 BUILDING MODELS

In our system, the buildings are represented by B-rep as described in section 1, with a slight adaptation: B-rep as described in section 1 can only be used for simple buildings not containing surfaces having holes (inner boundaries). If this is not the case, edges have to be grouped to form closed polygons, each surface has to contain references to a set of these closed polygons ("loops"), and one of these loops has to be labelled to be the outer boundary of the surface. The order of the vertices of a loop defines which part of 3D space is "inside" and which is "outside".

Both parametric and prismatic models are provided in our knowledge base. Whereas the construction rules for the prismatic model have to be programmed, the parameterised models contained in the knowledge base are data, so that the knowledge base can be adapted by the user. In order to add a new primitive to the knowledge base, the user has to provide a set of corner points v, a set of surfaces s and a set of edges e. For each edge, its starting and end vertices and the neighbouring surfaces have to be defined. For each surface, the mirror matrix $\mathbf{M}$ and either a subset of the coefficients a_{jk}, b_{ik}, c_{ij} from equation 2 defining its mathematical formulation or alternatively a reference to a symmetrical surface have to be provided. In addition, the outer boundary loop of the surface has to be defined by an ordered set of vertices. Note that all surfaces are formulated in the same observation co-ordinate system which, in addition, is only rotated around the Z axis. Thus, in all surface equations, the same exterior reference point $\mathbf{P}_0$ and the same rotations θ are used, and of the three rotation angles of θ only one angle κ is required. Taking the saddle back roof as an example, the following formulation would be adequate (figure 2): The reference point $\mathbf{P}_0$ is situated in the centre of the floor. Its object co-ordinates determine the position of the building in object space. The facades are vertical, thus only one rotation κ around the Z-axis is required to determine the orientation of the building in object space. The model consists of 7 surfaces:

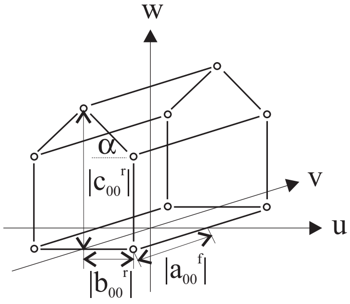

Figure 2: Parameterisation of a building model.

1. Floor: $w_0 = 0$
2. Front facade: $v_0 = a_{00}^f$
3. Back facade: $v_0 = a_{00}^f$, $m_v = -1$: the same parameters as (2), but mirrored with respect to the uw-plane.
4. Right facade: $u_0 = b_{00}^r$
5. Left facade: $u_0 = b_{00}^r$, $m_u = -1$ the same parameters as (4), but mirrored with respect to the vw-plane.
6. Right roof plane : $w_0 = c_{00}^r + c_{10}^r \cdot u_R$
7. Left roof plane: $w_0 = c_{00}^r + c_{10}^r \cdot u_R$, $m_u = -1$: the same parameters as (6), but mirrored by the vw-plane.

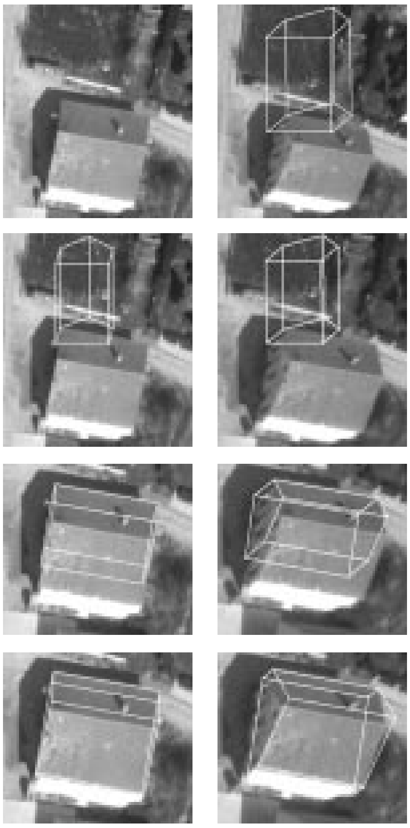

Figure 3: Determination of approximations.

When the user selects a primitive from the knowledge base, the data describing the geometrical properties are added to the data base of the adjustment system (section 2): the vertices and the reference point $\mathbf{P}_0$ are added to the reference system, a new set of rotations θ is created and two angles declared constant, for each surface an observation room of type "surface observation" is created and the vertices belonging to the surface are added to that room. In addition, for each surface having its own parameters a_{jk}, b_{ik}, c_{ij}, a new parameter set is created and the parameters contained in the description are added to it. With respect to surfaces being declared symmetrical, the elements of the mirror matrix have to be set according to the description and the reference pointing to the surface parameters has to be set to point to the symmetrical one. As each vertex is contained in three surfaces, its co-ordinates can always be computed from the intersection of these surfaces. The remaining parameters (three co-ordinates of $\mathbf{P}_0$, the building orientation κ and the surface parameters, in case of the saddle back roof a_{00}^f, b_{00}^r, c_{00}^r and c_{10}^r) describe the building. For these parameters, the user has to successively provide approximate values as described in section 4. As long as no such information is available, for each of the parameters, a parameter observation (section 2.1.3) is provided in order to make that parameter determinable in adjustment.

Although the building primitive is formulated in B-rep, it is determined by a minimum set of parameters which, in addition, can be interpreted easily: the building length $l = 2 \cdot |a_{00}^f|$, the building width $w = 2 \cdot |b_{00}^r|$, the building height $h = |c_{00}|$, and for the obliquity angle α of the roof we get $tan(\alpha) = |c_{10}|$. Thus, using the way of formulation described in section 2, we get rid of the over-parameterisation usually associated with B-rep without losing the flexibility of that form of modelling.

4 INTERACTIVE DETERMINATION OF APPROXIMATE BUILDING PARAMETERS

As soon as a primitive has been selected from the knowledge base, approximations for the building parameters described in the previous section have to be determined interactively. Typically, two images will be open for visual determination of approximations, and a wire frame representation of the building will be superimposed to the images using the current building parameters (figure 3). The human operator can interactively identify building vertices by mouse clicks, which results in the two camera co-ordinates of the measured point to be inserted to the adjustment data base. After each of these user interactions, the building parameters have to be updated taking into account the new piece of information provided by the user and using default values for the parameters which can not yet be determined. In this context it is the main problem to find out which building parameter(s) can be determined by the new point. This problem is solved using robust hybrid adjustment of all the information available about the building at the given time: the image co-ordinates of the vertices provided by the user, the surface observations and the observations for the parameters described in the previous section, the unknowns being the building parameters and the co-ordinates of the vertices. Note that only contradicting observations obtain large residuals in adjustment: observations not being checked by other ones are required to determine a certain parameter, and their residuals will be close to zero. We use this behaviour to determine which observed parameters contradict the other observations by performing robust estimation as described in section 2, applying re-weighting only to the observed parameters. After a few iterations, the observed parameters contradicting the information provided by the user are detected to be gross errors, and they will be eliminated from adjustment.

Figure 3 shows an example for interactive determination of the parameters of a building. The upper row shows the situation after the first image point has been measured in the right image. By these two image co-ordinates, the reference point $\mathbf{P}_0$ can be placed along the image ray, thus its planimetric co-ordinates can be determined, but not its height. $\mathbf{P}_0$ will be placed along the image ray, its Z co-ordinate being still determined by the default observation. As that default is not a very good one in the example, the wire frame is outside the displayed region in the left image. After measuring the homologous point in the left image, that point can be determined by spatial intersection. $\mathbf{P}_0$ can now be determined in 3D space (second row in figure 3). Measuring a second point in one image will determine κ and one of the building parameters, in this case a_{00}^f (third row in figure 3). Note that in this case it is necessary to compute a better approximation for κ first because otherwise adjustment would not converge. Finally, after measuring a third point in one of the images, another parameter (b_{00}^r) can be determined (fourth row in figure 3). Note that this principle works independently of the order in which the operator measures the points. In addition, there are no restrictions with respect to the number of points which have to be measured interactively. However, in the example the model is already placed well enough so that the automated modules will work.

5 AUTOMATIC FINE MEASUREMENT

5.1 Hierarchical reconstruction of object surfaces

Automatic fine measurement is implemented in the framework for automatic object surface reconstruction described in detail in (Rottensteiner, 1998). We use hierarchical multi-image feature based matching for automatic fine measurement. From the digital images, image pyramids are computed, and matching is applied iteratively to all pyramid levels. The object to be reconstructed is the building model which has been initialised by approximate values as described in section 4. First, the approximated building model is projected into all images to determine regions of interest. In these regions of interest, salient image lines are extracted from each pyramid level using the framework described in (Fuchs, 1995). The edge pixel streaks we obtain from edge tracking are thinned out using a merging algorithm so that the image lines are finally represented by polygons having relatively long sides, the length depending on the actual shape of the line. These image lines have to be matched to the edges of our building models. Matching is first applied to the upper level of the image pyramids (the level having the coarsest spatial resolution). The resulting representation of the building is then used as an approximation for a reduction of search space in the next lower pyramid level, and so on. The process is terminated as soon as the lowest level of the image pyramids (i.e. the level with the highest spatial resolution) has been reached. The coarse-to-fine approach using image pyramids is necessary to improve the radius of convergence of the method: in the higher levels of the image pyramids, the approximate positions of the projected building edges are only a few pixels away from the actual image edges. In addition, by lowpass filtering, which is used to produce the image pyramids, only the most salient image edges will survive whereas in the lower pyramid levels, more details and noise influences are available which are candidates for false matches. We also want to emphasise the benefits of using more than two images for matching. Due to shadow effects and occlusions, there are always object lines which are not visible in one or more images. Using a greater number of images helps to overcome that problem.

5.2 Generation and evaluation of correspondence hypotheses

The matching process on a given pyramid level itself consists of two phases: First, hypotheses for correspondence between image and object lines have to be generated based on the approximate position of the object lines. These initial hypotheses still contain errors which have to be eliminated in the second step, the evaluation of these hypotheses with respect to their consistency with the building model (Rottensteiner, 1998).

Hypotheses generation comprises the assignment of extracted image lines to building edges. Each side of the extracted image polygons is a straight line segment which might be a match for a given building edge or not. The building edges are projected to all images, and the straight line segments are compared to the projected object edges: A straight line segment is considered to be a match if the distance of both end points from the object line is smaller than a certain threshold, e.g. 2 pixels, and if both image and object lines are approximately parallel. If a straight line segment has been found to be a match of an object line in a certain image, the straight line segment is densified using an average point distance, e.g. 10 pixels (figure 4). After that, both the start and end points and the points inserted for densification are added to the adjustment data base: each point is added to the image the line segment belongs to and to both the object surfaces the object edge is neighbour of. This means that we get 4 observations for each image point assigned to an object line, i.e. two camera co-ordinates and two surface observations, whereas three new unknowns (the point's object co-ordinates) have to be determined. Thus, each line point added to the adjustment data base increases redundancy by 1. The longer a matched line segment is, the more observations it will add to adjustment and the more it will influence the results. However, the matching procedure is applied to all object edges in all images. Thus, we obtain an enormous redundancy which still renders possible the elimination of false matches. Using a correct stochastic model of adjustment is very important in this context. The surface observations are given a relatively high weight ($\sigma_{surf} = \pm 1$cm), and the weight of the image observation is computed from an estimate of $\sigma_{img} = \pm 0.5$ pixel.

Robust estimation as described in section 2.2 is applied for hypotheses verification, the re-weighting scheme being applied to the image observations only. For the time being, one observation is eliminated at each adjustment iteration. All the observations inserted to the adjustment data base are adjusted simultaneously, the unknowns being the building parameters, the building vertices and the object co-ordinates of the inserted line points. The relations between the object surfaces are enforced by their formulation (section 3). Thus, the edges are not adjusted separately, but together, and they all influence each other. The number of observations used to determine one single building may be high (up to several thousand), and the number of outliers is kept small by hierarchical processing as described in the previous section. An example for automatic fine measurement will be given in section 6.

6 EXPERIMENTS

The example presented in this section comes from a test project which is currently carried out in the village of Stoitzendorf in Lower Austria. A small aerial block of two strips (image scale: 1:3500, camera constant: 150 mm) with 80% overlap and 60% side lap was flown there. The images were scanned at a resolution of 15 μm. The configuration of the block guarantees each part of the village to be visible in at least four images. Most buildings are even visible in six images, as it is the case with our example building.

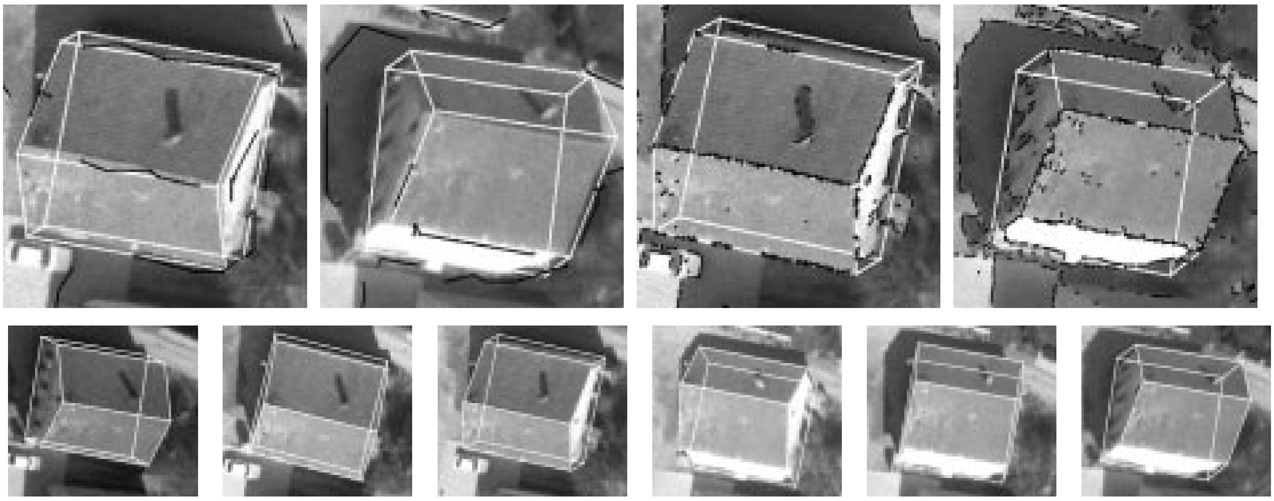

Figure 4: Automatical fine measurement. Upper row: extracted image lines (black lines), points inserted to the adjustment data base (black crosses) and matching results (white). Left two images: pyramid level 3 (120 μm), right two images: pyramid level 0 (15 μm). Second row: final position of the building model in all six images (white).

Figure 4 shows the matching process for one of the buildings in our test project. The numerical values of the building parameters are contained in table 1. It is the same building for which the determination of approximations was depicted in figure 3, where the initial state for two of the six images can be seen in the last row. Note that the eaves are already positioned relatively well, whereas the approximations for the roof tilt and the height of the central ridge are rather coarse (figure 3, last row). Matching started at level 3 of the image pyramids (120 μm). The distance threshold for matching was chosen to be 2 pixels, thus the radius of convergence is actually $2 \cdot 2^3 = 16$ pixels of the original resolution. Due to the lowpass filtering, only few matching lines are found, but there is enough information available in all six images so that the procedure converges (e.g. the left two images in upper row in figure 4, second row in table 1). Matching is iteratively applied to all pyramid levels until the final resolution is reached (level 0, e.g. the right two images in upper row in figure 4). Note that at this resolution many lines not being relevant four building reconstruction are extracted in the images which would be candidates for false matches if the coarse-to-fine strategy described in section 5.1 were not applied. Finally, in the second row of figure 4, the final matching result is superimposed to all six images.

We have also determined the parameters of that building by manual measurement of the building corners in the images. Comparing the last two lines of table 1, we see that the results of manual measurement differ slightly (up to 10 cm) from the matching results in level 0. The difference corresponds to about 2 pixels in the images and can partially be explained by the uncertainty of definition of the building edges: the matching process determines the vertices by intersection of long edges. In some cases, it might take the edges of the gutters rather than the edges of the roof. However, this is a problem of interpretation rather than one of accuracy. The internal accuracy of the building parameters was estimated to be about 1 to 2 cm, which is caused by the great number of observations taking part in adjustment (about 4600) leading to a great redundancy (1102). Note that in the current implementation, we apply matching only to roof edges. That is why parameter c_{00} corresponding to the height of the central ridge of the roof is constant in table 1. The floor could be positioned either by measuring one terrain point in the vicinity of the building and assigning its height to the floor or by intersecting the wall edges with a digital terrain model if such a model were available.

State	Resolution	X_0 [m]	Y_0 [m]	Z_0 [m]	κ [gon]	a^f_{00} [m]	b^r_{00} [m]	c^r_{00} [m]	c^r_{10} [%]
Initial	-	-247.32	180.47	273.90	289.09	6.26	-6,79	-7.00	-77.85
Level 3	120 μm	-247.17	180.36	273.07	291.33	6.76	-6,67	-7.00	-61.58
Level 2	60 μm	-247.04	180.39	272.26	289.87	6.66	-6,73	-7.00	-53.56
Level 1	30 μm	-247.15	180.48	271.95	289.34	6.64	-6,75	-7.00	-49.27
Level 0	15 μm	-247.21	180.53	271.90	289.25	6.64	-6,75	-7.00	-48.56
Manual	15 μm	-247.30	180.57	272.01	289.25	6.59	-6,75	-7.00	-51.31

Table 1: Results of the matching process for the building in figure 4.

7 CONCLUSIONS

In this paper we have described a new system for semi-automatic building extraction. The system is embedded in the photogrammetric plotting system ORPHEUS which offers a graphics user interface for the adjustment system ORIENT as well as modules for visualisation of and interactive measurement in digital images. The core of the system is a new approach for modelling building primitives based on ORIENT's possibilities for the formulation of surface observations and a close integration of hybrid adjustment which supports both the interactive tools and the automated modules of the system. The interactive tools turned out to be quite efficient because few user interactions are required to position the building model in the images. Note that even if the automatic determination of the building parameters fails, the work flow of building extraction is sped up significantly by providing the knowledge base of building primitives. Automatic fine measurement exists in a prototype program which gives the very promising results presented in this paper. However, the program still has ample possibilities for improvement of its operability. The process can be sped up by eliminating the superfluous unknowns (the object co-ordinates of the points inserted in the matching phase) from the normal equation system and by reducing the number of iterations required for robust estimation by investigating other strategies for re-weighting than the one described in section 2.2. The evaluation of the performance of automatic fine measurement will be carried out in a major project from which the example described in this paper was taken.

ACKNOWLEDGEMENT

This work was funded by the Austrian Science Fund in the framework of the Research Program S7004-MAT on Pattern Recognition and Digital Image Processing.

REFERENCES

Brunn, A., 1998. Techniques for Automatic Building Extraction. In: Third Course in Digital Photogrammetry, Institute for Photogrammetry at Bonn University and Landesvermessungsamt Nordrhein-Westfalen, Bonn, Germany.

Fuchs, C., 1995. Feature Extraction. In: Second Course in Digital Photogrammetry, Institute for Photogrammetry at Bonn University and Landesvermessungsamt Nordrhein-Westfalen, Bonn, Germany.

Gülch, E., Müller, H., Läbe, T. and Ragia, L., 1998. On the Performance of Semi-automatic Building Extraction. International Archives of Photogrammetry and Remote Sensing, Vol. XXXII-3/1, Columbus, OH, pp. 331–338.

Kager, H., 1989. ORIENT: A Universal Photogrammetric Adjustment System. In: A. Grün and H. Kahmen (eds), Optical 3-D Measurement, Herbert Wichmann Verlag, Karlsruhe, Germany, pp. 447–455.

Kraus, K., 1997. Photogrammetry Volume 2. Advanced Methods and Applications. fourth edn, Dümmler Verlag, Bonn, Germany. With contributions by J. Jansa and H. Kager.

Müller, H., 1998. Experiences with Semiautomatic Building Extraction. In: Third Course in Digital Photogrammetry, Institute for Photogrammetry at Bonn University and Landesvermessungsamt Nordrhein-Westfalen, Bonn, Germany.

Rottensteiner, F., 1998. Object Reconstruction in a Bundle Block Environment. International Archives of Photogrammetry and Remote Sensing, Vol. XXXII-3/1, Columbus, OH, pp. 177–183.

Veldhuis, H., 1998. Performance Analysis of two Fitting Algorithms for the Measurement of Parametrised Objects. International Archives of Photogrammetry and Remote Sensing, Vol. XXXII-3/1, Columbus, OH, pp. 400–408.