ORTHO-MOSAICS AND DIGITAL ELEVATION MODELS FROM AIRBORNE VIDEO IMAGERY USING PARALLEL GLOBAL OBJECT RECONSTRUCTION

Mikael Holm, George Denissoff, Kaj Juslin, Matti Paljakka, Markku Rantasuo, Susanna Rautakorpi
VTT Automation, Finland
E-mail: Mikael.Holm@vtt.fi

Commision III, Working Group 2

KEY WORDS: DEM/DTM, Orthoimage, Mosaic, Video, GPS, Global Matching, Object Reconstruction, Parallel Computing

ABSTRACT

One of the main obstacles for the use of video camera in airborne mapping applications is the small field of view compared to the resolution of the video camera. Single video images cover only a small part of the area to be mapped. Therefore automated methods to combine huge amounts of video images into single image-mosaics are essential.

In this paper a system under development is described, which will take as input thousands of aerial video images and will output a digital ortho-mosaic and also a digital elevation model of the covered area. The "3D-image-mosaic" will be made automatically using the methods of global matching or global object reconstruction. In this case an object based approach is used in the matching. As this kind of methods are mostly very computation intensive parallel computation is used.

First, the ideas behind the system are described, including the use of GPS navigation data with the on-line digitization of the images on the aircraft. Then, the present status of the development of the matching software is described, including breakline detection and solving of the normal equations on a parallel computer.

1 INTRODUCTION

The effective use of airborne video imagery in mapping or environmental monitoring tasks requires the use of automatic methods to combine thousands of images into one or a few image-mosaics. To make accurate mosaics a digital elevation model (DEM) of the terrain is needed. As a DEM of the area to be mapped in many cases does not exist, a system creating the mosaics should be able to compute the DEM itself, using the image data.

One algorithm capable of computing image mosaics and DEM's from image data is global object based multi-image matching, also called global object reconstruction. It is a general model for digital photogrammetry, integrating area-based multi-image matching, point determination, object surface reconstruction and orthoimage generation. Using this model the unknown quantities are estimated directly from the pixel intensity values and from control information in a nonlinear least squares adjustment. The unknown quantities are the geometric and radiometric parameters of the approximation of the object surface (*e.g.* the heights of a DEM and the brightness values of each point on the surface), and the orientation parameters of the images. Any desired number of images, scanned in various spectral bands, can be processed simultaneously.

This algorithm is a generalisation of the least squares matching methods (Ebner *et al.*, 1987, Ebner & Heipke 1988). Similar concepts have been developed independently (Wrobel 1987, Helava 1988, Weisensee 1992). A detailed description of this matching algorithm and an evaluation using synthetic aerial and real close range imagery can be found in (Heipke 1990, Heipke 1992). The first controlled tests of the approach using real aerial images can be found in (Ebner *et al.*, 1993, Holm 1994).

In this paper the present status of the ESPRIT-III GLORE projective is described, where the object is to demonstrate the capabilities of the method for the creation of massive mosaics of digitized aerial video imagery, and to create a prototype system consisting of parallel hardware and software for such usage. The work is funded by the Commission of the European Communities.

The prototype system under development consists of two parts. The first part is on the aircraft taking care of the digitizing of the video images. The second part is on ground taking care of the orthoimage mosaic and DEM generation. The airborne part is described in section 2. In section 3 the matching method is shortly described. In section 4 the implementation on parallel hardware is described. The most computation intensive part of the system is the solving of the normal equations. The description of the methods used can be found in section 5. In order to reduce the number of unknowns and thus speed up the computation, irregular DEM's are used. This is done using breakline detection as described in section 6.

2 THE AIRBORNE DIGITIZATION OF VIDEO IMAGERY

The video images are captured and digitized onto the hard-disk of a PC on board the aircraft during the flight. The Super-VHS video camera is connected directly to the computer. Traditional tape recorders are not used. The intervals between the grabbing of the video frames, image brightness etc. can be adjusted interactively using the PC and a piece of software developed for this purpose. The grabbed frames are stored as 24-bit truecolor images.

While the PC is digitizing video frames a laptop is gathering navigation information using Real time Differential Global Positioning System (RDGPS) measurements. The laptop is equipped with a DGPS card and Radio Data System (RDS) receiver. The differential corrections for the GPS measurements are carried out using the RDS coded correction information sent by the national broadcasting company of Finland. As a very cheap and simple system is used the accuracy of the RDGPS coordinates — received once every second — are in the order of 3 – 5 m. The laptop shows the planned flight route and the real flight route graphically in real time, as well as the metrical deviations from the planned route. Using this equipment the pilot can easily keep on the track horizontally and vertically, and maintain the planned flying speed relative to the ground.

The digitizing computer is connected to the RDGPS card of the laptop. The clock of the digitizing computer is synchronised with the GPS signal using a synchronisation card. The coordinates of the position of the aircraft during the last few seconds prior to the digitization are stored with corresponding time stamps in the header of each image. Using this information attached to the images the position of the camera during the frame grabbing can be estimated with an accuracy of 3 – 7 metres. These coordinates are used as initial values in the global matching, where more accurate coordinates are computed. The in-flight system is shown in figure 1. Missing from the image are the tilt-sensors giving the approximate attitude information for each video-image.

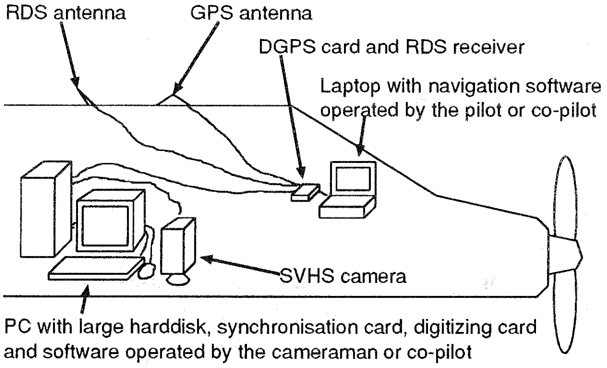
RDS antenna
GPS antenna
DGPS card and RDS receiver
Laptop with navigation software operated by the pilot or co-pilot
SVHS camera
PC with large harddisk, synchronisation card, digitizing card and software operated by the cameraman or co-pilot

Figure 1: The on-board digitizing and navigation system

3 DESCRIPTION OF THE MATCHING APPROACH

First, a geometric and a radiometric model in object space are introduced (Figure 2). The geometric model consists of a grid DEM. The grid is defined in the XY-plane of the object surface with grid nodes X_k, Y_l and grid heights $Z(X_k, Y_l) = Z_{k,l}$. The mesh size depends on the roughness of the terrain. A height $Z(X, Y)$ at an arbitrary point is interpolated from the neighbouring grid heights. In the radiometric model object surface elements of constant size are defined within each grid mesh. The size is chosen approximately equal to the pixel size multiplied by the average image scale factor. An object intensity value $G(X, Y)$ is assigned to each object surface element. The centre P of each object surface element is projected into the different images using the collinearity equations. Subsequently image intensity values g at the corresponding locations x, y in pixel space can be resampled from the original pixel intensity values.

As the assumptions of constant illumination parameters and perfect Lambertian reflection are not rigorously met in the imaging process, a radiometric image transformation T is introduced to compensate at least partially for the deviations. This simplification does not hold in general, but all image matching algorithms without prior knowledge about the object surface reflectance properties have the same problem.

In the following, the grid heights $Z_{k,l}$, the parameters p for the exterior orientation of the images, the intensity values $G(X, Y)$ of the object surface elements, and the parameters of the radiometric transformation T are treated as unknowns. They are estimated directly from the observations $g(x,y)$ and control information in a least squares adjustment. Thus, $g(x,y)$ depends on $Z_{k,l}$ and p. For each object surface element, as many values $g(x,y)$ can be computed as there are images, and as many observation equations of the following type can be formulated:

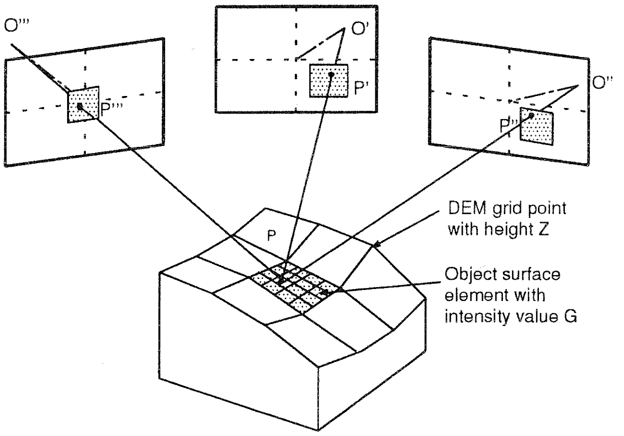
O'''
O'
O''
P'''
P'
P''
P
DEM grid point with height Z
Object surface element with intensity value G

Figure 2: The geometric and radiometric models, and the transformation from object to image space

$$v = G - T[g(x(Z,p), y(Z,p))], \qquad (1)$$

where

v	is the residual of the observation $T[g]$
G	is the unknown intensity value of the object surface element
T	is the radiometric transformation
g	is the resampled image intensity value
x, y	are the pixel coordinates
Z	are the unknown heights of the surrounding grid points
p	are the unknown parameters for the image orientations

The system of observation equations is completed by introducing control information with appropriate standard deviations. Since the observation equations are nonlinear in Z and p, the solution of the least squares adjustment is found iteratively.

4 IMPLEMENTATION ON PARALLEL HARDWARE

In order to enable the operational usage of the method within practical computation times the global object reconstruction algorithm is being implemented as a massively parallel MIMD (multiple instructions - multiple data) computing application. Due to the complexities resulting from parallelization characterized by scores of tasks interacting with numerous message types, a need to model and design the application with a uniform, structured and formal notation soon became self-evident. The Object Modelling Technique (OMT) (Rumbaugh *et al.* 1991) was chosen for this purpose because it was seen to be both a powerful general-purpose modelling technique and particularly well-suited for design of parallel applications.

In spite of massive parallelization the application has been designed for portability. This is being achieved by coding the application with the ANSI-C-language and using only the widely available public-domain Parallel Virtual Machine (PVM) library (Geist & Beguelin 1994) as a tool for parallelization. The application has also been designed to minimize both the memory requirements and the communication between the nodes of the parallel computer, which will also strengthen the portability of the application.

The application is currently being developed and tested on a TN330 parallel computer manufactured by Telmat Multinode, France. The TN330 being used by the project has 68 nodes with 32 Mbytes of memory. Each node consist of a superscalar T9000 processor capable of peak performance of 200 MIPS and 25 MFLOPS. Each T9000 processor has four communication links which provide a total of 80 Mbytes/second bidirectional bandwith.

In the least squares adjustment the size and structure of the normal equations matrix depend on the unknown quantities. Even if the brightness values of each point on the surface are reduced from the normal equations there still are millions of unknowns in the system. The magnitude of the system implies a demand for much memory and fast computing. This problem is tackled in four ways:

1. the unknown intensity values G in (1) are eliminated in the matching,
2. instead of using a regular DEM grid a variable sized grid network following breaklines will be used to cut down the number of unknowns (c.f. section 6),
3. optimizing the solving of the normal equations (c.f. section 5), and
4. the system is divided into smaller geographical parts and processed in parallel.

5 THE SOLUTION OF THE NORMAL EQUATIONS

The most computation intensive part of the system is the solution of the normal equations:

$$Nx = h$$

Here, the matrix N of the normal equations is obtained from the design matrix A and the diagonal weight matrix W.

$$N = A^T WA, \quad h = A^T Wl$$

The system includes two models and their mutual dependencies. One of the models represent the DEM grid points, each dependent on all eight grid points around it. The other one of the two models represent the orientation parameters of each image. The matrix N can be dissected into blocks corresponding to the two models:

$$N = \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix}$$

Here, the submatrix N_{11} corresponds to the DEM, which implies that its dimension is equal to the number of grid points and that there are nine non-zero elements on each row (if only regular DEM's are used). If the matrix N is formed by sweeping through the DEM row by row, a diagonal band with a semi-bandwidth of one and two bands further away from the diagonal symmetrically around it are obtained.

The submatrix N_{22} corresponds to the image orientations, which gives the submatrix a dimension of $6 \cdot n_{images}$. Since the orientation parameters of different pictures are not dependent on each other, N_{22} consists of blocks of size 6×6. In the calculations for storage and operation count, the submatrix is assumed banded with a semi-bandwidth of 5.

Since the system is overdetermined, the matrix A is a full rank matrix and the normal equations' matrix N is positive definite, which implies that there are a number of methods to solve the equation $Nx = h$.

5.1 Solution algorithms

Because the matrix N is large, solution methods which do not require a great deal of memory are preferred. In general, N is stored in a sparse matrix format, and whenever possible, the symmetry of N is exploited by storing only the upper triangular of N.

The operation counts of various solution methods can be estimated using formulae given in (Pissanetzky 1984). For the factorization of a banded matrix, it gives:

$\frac{1}{2}\beta(\beta+3)n - \frac{1}{3}\beta^3 - \beta^2 - \frac{2}{3}\beta$ multiplications and divisions
$\frac{1}{2}\beta(\beta+1)n - \frac{1}{3}\beta^3 - \frac{1}{2}\beta^2 - \frac{1}{6}\beta$ additions

Here, n is the dimension and β is the semi-bandwidth of the matrix. Similarly, the following formulae are given for the the forward and back substitution of a banded matrix:

$(2\beta+1)n - \beta^2 - \beta$ multiplications and divisions
$2\beta n - \beta^2 - \beta$ additions

Direct solution by Cholesky block factorization. Out of the direct solution methods, Cholesky block factorization is a very beneficial method in this case (George & Liu 1981, Pissanetzky 1984). In that method, the matrix is dissected into blocks and thus subsystems which are then factorized independently. The mutual dependencies of the diagonal blocks are moved to the far right in the submatrix N_{11} and thus joined to the block N_{12}, which is not actually factorized but otherwise taken into account in the solution of the lower part of the equation.

Several grid ordering schemes: the Reverse Cuthill - McKee scheme, the Minimum Degree algorithm and the Nested Dissection algorithm have been suggested for the reduction of matrix fill-in in the factorization phase. Fill-in reduction has two positive effects: on one hand, memory is saved, and on the other hand, both the factorization and the forward and back substitutions demand fewer operations than if no specific ordering scheme were used.

The direct solution is parallelized by spawning two tasks on every node and sending a slice of N_{22} to one of them and a slice of both N_{11} and N_{12} to the other one. For symmetry reasons, the block N_{21} is not stored at all, and only the upper triangular of the block N_{11} is stored. The whole block N_{11} is stored, which makes it possible to perform an efficient parallelized Gauss elimination on it. The blocks N_{11} and N_{12} are stored in sets of linked lists to save memory in the factorization process. In the block factorization, the block N_{22} becomes essentially full, and it is thus stored in an array.

For a Gauss elimination on p processors, the following operation counts can be obtained:

$\frac{n^3}{p} + (\frac{1}{p} + \frac{1}{2})n^2 + (\frac{3}{2} - \frac{1}{p})n$ multiplications and divisions
$\frac{n^3}{p} - \frac{3n^2}{2p} - \frac{n}{2p}$ additions

In the following, the blocks N_{11} and N_{22} are given dimensions n_1 and n_2 respectively. The block N_{11} is assumed banded with a semi-bandwidth of $\beta = \sqrt{\frac{n_1}{p}}$, which is a worst-case approximation. Due to the block formation, a growth of $2(\sqrt{p}-1)\sqrt{n_1}$ is expected in the dimension n_2. The operation count for the direct solution is hence proportional to the square of n_1 and cube of n_2. If the band of N_{11} becomes full, the memory required by every node is in the direct solution approximately:

$$\frac{n_1-2(\sqrt{p}-1)\sqrt{n_1}}{p}\cdot(\sqrt{\frac{n_1-2(\sqrt{p}-1)\sqrt{n_1}}{p}}+39.5)\cdot 16\,\text{byte}$$
$$+\frac{n_2+2(\sqrt{p}-1)\sqrt{n_1}}{p}\cdot 8\,\text{byte}$$

Iterative solution by point-Jacobi iteration. Also an iterative method has been developed for this application. In the outer iteration loop of that iteration, the diagonal blocks N_{11} and N_{22} are inverted and the effects of the blocks N_{12} and N_{21} are taken into account by an iteration. In an inner iteration loop, the inversion of the block N_{11} has been replaced by an iteration where only the diagonal band of the submatrix is factorized.

The iterative solution is parallelized for a distributed system by storing the block N_{22} in one node and sending a slice of both N_{11} and N_{12} to each of the rest of the computation nodes. Like in the direct solution, the block N_{21} is not stored at all, and only the upper half of the band including the diagonal of N_{11} is stored. For this method, the rest of the block N_{11} is stored in sets of linked lists which contain the both triangulars of the block. Only the upper half of the band of N_{22} is stored, and similarly as above, the block N_{12} is stored in a set of linked lists.

According to the same formulae as above, the preparing for the iteration *i.e.* factorizing the bands of the blocks N_{11} and N_{22} involves a number of operations linearly proportional to the system dimension. Similarly, the operation counts of both the outer and the inner loop are linearly dependent on the system size.

The memory required by the iterative algorithm is $n_2 \cdot 48\,\text{byte}$ for the node responsible for N_{22} and $\frac{n_1}{p-1}\cdot 776\,\text{byte}$ for the nodes responsible for N_{11}.

In this iteration, the two models included in the matrix are solved independently and the dependencies between the models are solved iteratively. Another possible iteration not yet implemented would solve both the terrain and the orientation models on various subareas independently and take the subarea boundaries into account by an iteration.

5.2 Test results on the normal equations' solver.

Only very preliminary tests have been performed, but a few fundamental results have been achieved.

The direct solution has appeared too slow: on any row ordering scheme, a system of 3475 equations which was solved iteratively in six seconds, was solved by Cholesky block factorization in no less than 40 minutes on one computation node. The solution time by the direct solution is awaited to be proportional to the square of the system dimension, and thus solving a system of million equations on 68 transputers would take several years by that method.

The iterative solution has proved significantly faster: the solution of a million equations on 68 nodes could probably take far less than an hour, which can be well coped with.

Probably the future solution method will be a combination of the two: the direct solution is used for obtaining the initial values for the iterative solution.

6 DETECTION AND MODELLING OF BREAKLINES

When developing the breakline detection algorithm, emphasis was put on handling of large areas when the number of unknown DEM points becomes large. The main idea was to lower the number of unknown DEM points by detecting the locations were breaklines exist and modelling breaklines with a denser grid on those places and using a sparser network elsewhere. Possibilities of using triangles or squares as elements in the denser parts of the DEM were studied by implementing some general edge detection algorithms for breakline detection from difference orthoimages.

6.1 Detection of breaklines from orthoimages

Breaklines can be regarded as gross errors of the mathematical model if the object surface is modelled without considering breaklines, and smoothness constraints are used for reconstruction of the smooth object surface. Detection of the defects in the mathematical model by using difference orthoimages was proposed, *e.g.*, in (Ebner *et al.*, 1993) for breakline detection and in (Hahn 1992) for automatic digital terrain model verification. The main idea of this kind of residual analysis method is to compute one orthoimage per input image. The difference images between the input images should be identical, if the geometric and the radiometric part of the mathematical model were correct. Large differences between the orthoimages can therefore be caused by breaklines.

The orthoimages and difference orthoimages (Figure 3) were calculated in two different pyramid levels. The scale of the original aerial image was approximately 1 : 15 000 and the heights of the 320 × 320 m^2 test area varied from 5 to 27 metres. Several edge detection operators, such as the Sobel, Prewitt, Kirsch, Robinson operators and a symmetric exponential filter of infinite size (ISEF) (Shen & Castan 1992) were implemented for breakline detection from orthoimages in different pyramid levels. Due to noisy orthoimages median filtering was used before using the mask operators, while in the case of the ISEF-operator image filtering and edge detection was performed at the same time. The different masks operators gave results of the same kind. However, the results of the ISEF-operator seemed to be better, *e.g.*, more continuous breaklines were obtained. The location of the most significant breakline corresponding to a creek on the left hand side of the image could be found more clearly in difference orthoimages than in the left and right orthoimages, while the road on the right corresponding to the other elongated but less significant breakline could be distinguished more clearly from the left and right orthoimages (Figure 3). After thresholding of the edge images an edge linking method was needed.

6.2 Developing a method for breakline modelling

Breaklines on the object surface can be very complicated and therefore only one of the simple cases was handled. First, it was assumed that there is only one (or none) breakline within a mesh and it passes through the whole mesh and secondly, an assumption about the smoothness of breaklines was made. Under these assumptions breaklines can be approximated by a line segment. Because the Hough transform is suitable for detection of straight lines, the suitability of the Hough transform for linking the edge pixels within a single mesh was tested. The goals of linking step were to study:

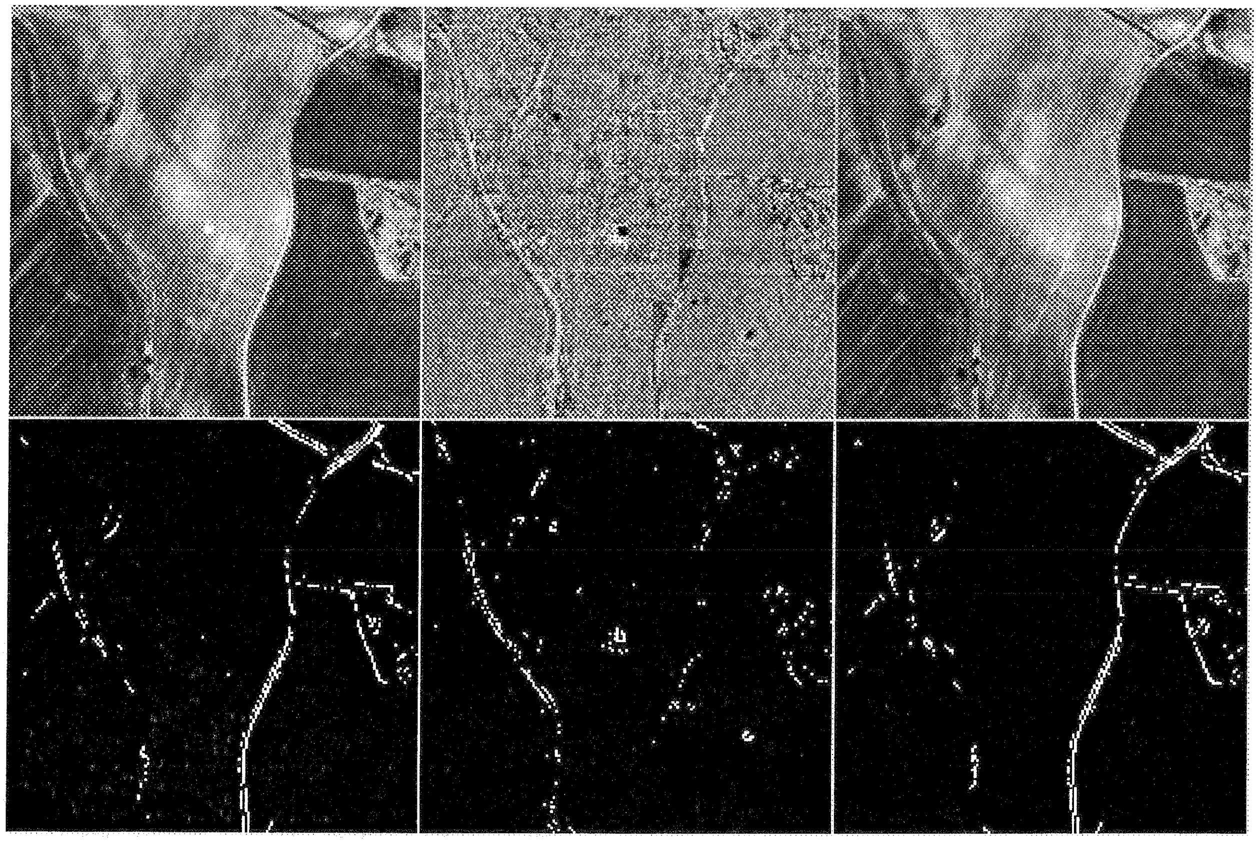

Figure 3: Above: left, difference and right orthoimage. Below: breakline detection from images by using ISEF-edge detector.

- whether the existence of a breakline within a single mesh can be determined by using the Hough transform,
- whether a possible breakline within a mesh can be determined approximately, and
- whether possible breaklines in neighbouring meshes could be modelled by using a single continuous line and triangles, or would it be easier to model breaklines by densifying the network with squares.

The Hough transform was computed for each mesh separately, for example, 16 × 16 Hough transforms were computed at pyramid level 2 for areas each consisting of 22 × 22 pixels. If the count of the global maximum cell in the accumulator array exceeded a specific threshold, it indicated that there was a breakline within that mesh. The threshold was used in order to filter out weak, insignificant breaklines and noise. The results of the Hough transform showed that the transform was sensitive to mesh partition. Due to noisy test images gaps appeared at breakline locations in which there were only a few edge pixels left within a single mesh, when a high threshold was used (c.f. the gray meshes between the white meshes in the left hand side of figure 4.a), while by using a lower threshold a lot of insignificant breaklines were found and the requirement to keep the number of new meshes low could not be fulfilled. To handle this problem a new algorithm for using two thresholds was developed. A higher threshold was used for edge points corresponding to significant breaklines and another lower threshold for potential, weak breaklines. The main idea was to get more global information about locations of breaklines by comparing the results of neighbouring meshes. If a chain of significant breakline meshes was found by using the higher threshold, it was assumed to indicate that neighbouring meshes corresponding to potential breaklines were also significant breakline meshes.

The results obtained by using the new algorithm seemed to be promising when the available test images were used. Almost all meshes corresponding to the most significant breakline were found and rough locations of the two most significant breaklines could be found, c.f. locations A and B in figure 4.c. Only a few insignificant breaklines, such as locations C, F and G were found. However, local disturbances, such as dust at locations D and E, could not be separated from breaklines when two orthoimages were used. When more images are used this should not be a problem.

On the basis of these test results, the new algorithm will be implemented to applicable parts to the program package for airborne video imagery being developed. By means of the algorithm it could be determined whether there was a breakline or not within a single mesh – at least in the most cases of the available test images – but finding the approximate locations of breaklines within single meshes seemed to be a more complicated problem. If the breaklines were modelled by using a continuous line that consists of line segments in adjacent meshes, at least two problems should be solved: connecting successive disconnected line segments, when different edges corresponding to the same breakline are found, and handling of the breakpoints or discontinuities that did not form a breakline, but rather form an area consisting of single breakpoints and causing occasional direction for corresponding line segment. Due to these problems a variable sized grid network utilizing rough information about the locations of the breakline meshes was found to be more

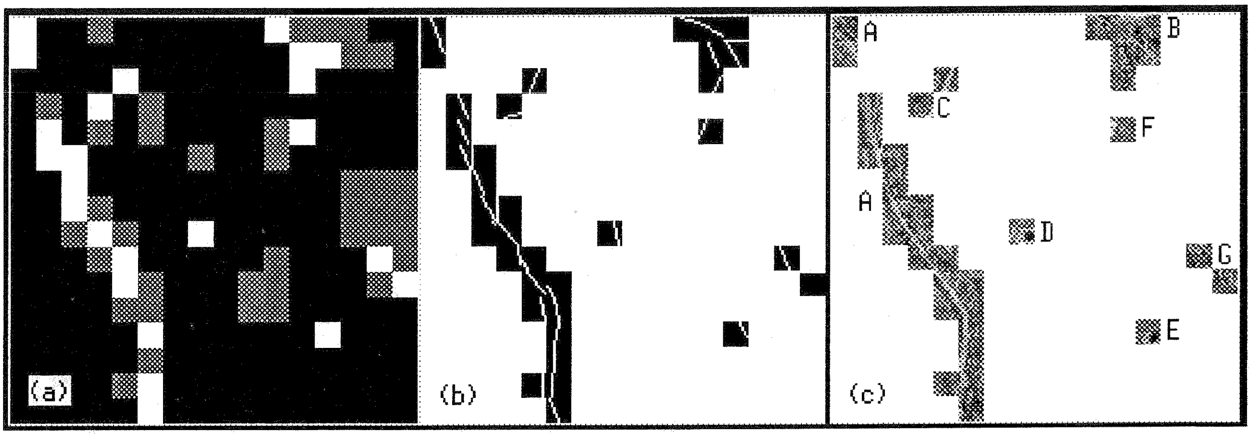

Figure 4: An example of the results of the new algorithm: (a) Situation after performing the Hough transform with two thresholds. Breakline meshes are illustrated in white, the potential breakline meshes in gray and non-breakline meshes in black. (b-c) Results of the whole algorithm. Non-breakline meshes are shown in white. Line segments approximating locations of breaklines within meshes are shown in (b) and the corresponding orthoimage in (c). See text for explanations of the labels.

suitable for breakline modelling than modelling breaklines with continuous line segments forming edges of triangles.

7 SUMMARY

In this paper an video image data acquisition system for mapping applications was presented. The acquisition system, still under development, consists of two parts:

1. First, a huge amount of true colour video images are captured from an aircraft. GPS-coordinates of the camera are included in the image headers, and are also used by the pilot in navigation via the real-time plot of the planned and real flight routes.
2. The digital images are transfered from the digitizing PC to a UNIX workstation, which is the host machine for a transputer based parallel computer. The system automatically creates a large orthoimage mosaic and a digital elevation model of the area.

The "3D-image-mosaic" will be made using the methods of global matching or global object reconstruction, a general model for digital photogrammetry, integrating area based multi-image matching, point determination, object surface reconstruction and orthoimage generation. As it is very computation intensive, a parallel computation is used.

REFERENCES

Ebner, H., D. Fritsch, W. Gillessen & C. Heipke 1987, Integration von Bildzuordnung und Objektrekonstruktion innerhalb der digitalen Photogrammetrie, Bildmessung und Luftbildwesen, 55, (5), 194–203.

Ebner, H. & C. Heipke 1988, Integration of digital image matching and object surface reconstruction, International Archives for Photogrammetry and Remote Sensing, 27 (B11/III), 534–545.

Ebner, H., C. Heipke & M. Holm 1993. Global image matching and surface reconstruction in object space using aerial images. In: Barrett, E.B., McKeown, D.M. Jr (eds.) Integrating photogrammetric techniques with scene analysis and machine vision. Proceedings of SPIE conference, Orlando, Florida, 14-15 April 1993. Vol. 1944, pp. 44–57.

Geist, A. A. Beguelin 1994. PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Networked Parallel Computing, The MIT Press.

George & Liu 1981. Computer Solution of Large Sparse Positive Definite Systems, Prentice - Hall, Englewood Cliffs, 1981

Hahn, M. 1992. Towards Automatic DTM Verification Exploiting Stereo Orthophotos. International Archives for Photogrammetry and Remote Sensing, Vol. 29, Com. 3, B3, pp. 233–240.

Heipke, C. 1990, Integration von digitaler Bildzuordnung, Punktbestimmung, Oberflächenrekonstruktion und Orthoprojektion in der digitalen Photogrammetrie, Deutsche Geodätische Kommission, Reihe C, 366. München 1990.

Heipke, C. 1992, A global approach for least squares image matching and surface reconstruction in object space, Photogrammetric Engineering and Remote Sensing, 58 (3), pp. 317–323.

Helava, U. V. 1988, Object-space least-squares correlation, Photogrammetric Engineering and Remote Sensing, 54, (6), pp. 711–714.

Holm, M. 1994, Global image matching and surface reconstruction in object space using three aerial images. Proceedings of ISPRS Commission III Symposium: Spatial Information from Digital Photogrammetry and Computer Vision. 5.–9. September 1994, pp. 379–386.

Pissanetzky 1984. Sparse Matrix Technology, Academic Press Inc., London.

Rumbaugh J., M. Blaha *et al.* 1991. Object-Oriented Modeling and Design, Prentice-Hall.

Shen, J., S. Castan 1992. An Optimal Linear Operator for Step Edge Detection. CVGIP: Graphical models and image processing, Vol. 54. No. 2, pp. 112–133.

Weisensee M. 1992 Modelle und Algorithmen für das Facetten-Stereosehen, Deutsche Geodätische Kommission, Reihe C, 374.

Wrobel, B. 1987. Facets Stereo Vision (FAST Vision) – A new approach to computer vision and to digital photogrammetry, Proceedings of ISPRS Intercommission Conference on "Fast Processing of Photogrammetric Data", pp. 231–258.