

DISTA – a portable software solution for 3D-compilation of photogrammetric image blocks

Frank Boochs, Hartmut Müller, Markus Neifer

i3mainz, Holzstrasse 36, D-55116 Mainz, Germany

Univ. of Applied Sciences/Mainz

ABSTRACT

A photogrammetric evaluation system used for the precise determination of 3D-coordinates from blocks of large metric images will be presented. First, the motivation for the development is shown, which is placed in the field of processing tools for photogrammetric evaluation tasks. As the use and availability of metric images of digital type rapidly increases corresponding equipment for the measuring process is needed. Systems which have been developed up to now are either very special ones, founded on high end graphics workstations with an according pricing or simple ones with restricted measuring functionality. A new conception will be shown, avoiding special high end graphics hardware but providing a complete processing chain for all elementary photogrammetric tasks ranging from preparatory steps over the formation of image blocks up to the automatic and interactive 3D-evaluation within digital stereo models. The presented system is based on PC-hardware equipped with off the shelf graphics boards and uses an object oriented design. The specific needs of a flexible measuring system and the corresponding requirements which have to be met by the system are shown. Important aspects as modularity and hardware independence and their value for the solution are shown. The design of the software will be presented and first results with a prototype realised on a powerful PC-hardware configuration will be featured.

Keywords: OpenGL, wxWindows, Geographic Information Systems, Digital Photogrammetry, Photogrammetric Workstations, 3D-Measurements, Stereoscopy

1. INTRODUCTION

Digital images are increasingly playing an important role in the photogrammetric business. Their primary advantage is to permit a higher degree of automation within evaluation processes. They of course allow additionally to apply conventional actions onto images like manual stereoscopic measurements, for example. However, the number of images existing in a project and their size make it necessary to have an intelligent solution for the data handling and a powerful computer equipment allowing to effectively display the images during the different interactive evaluation steps. Therefore conventional photogrammetric workstations are based on specialized graphic hardware with corresponding special software solutions. This concept is accompanied with high costs, dependency on proprietary hardware and low flexibility.

These disadvantages might be avoided by use of powerful standard hardware together with an open and modular software structure. This allows to reduce hardware costs, to simplify and accelerate the process of development and to adjust the functionality to different applications. In addition, it might be of value to integrate existing CAD packages into such an evaluation environment.

Based on the aspects of simplicity, modularity and flexibility a software solution (DISTA = digital stereoscopic evaluation architecture) has been designed and developed. In the following, motivation and background for the proposed solution are outlined, the design of the system is sketched and some technical details are presented.

2. PHOTOGRAMMETRIC TOOLS

2.1. GENERAL

Since the end 70ies photogrammetric instruments are working under computer control and allow to generate digital object data. They are controlled by a human operator and meet all requirements for an efficient data collection system.

Actually, the technological steps towards purely digital systems are ongoing. Digital systems have the advantage to contribute to the collection process by some automatic evaluation tools exploiting the content of the digital data. However, their use for a complete 3D-data collection is restricted to objects being represented by a certain number of directly mounted targets. Here, adopted search algorithms can be applied allowing to detect the targets and to determine their positions in space. Under certain conditions even real-time applications are possible using integrated camera systems capturing the image data and automatically exploiting them by means of processes running on the integrated computer system (Imetric, 1999; Leica, 1998).

However, in general, purely digital systems need human control, at least for the collection of the object 3D-data but in most cases for the collection of tie points too. These systems have to handle several individual images at the same time, or image pairs during the 3D evaluation process respectively. Typically, image sizes may range from 10.000 by 10.000 pixel up to 30.000 by 30.000 pixel each. Accordingly the work load for the computer system is very high, because all those steps need to have dynamic access to the image data to allow for real-time image movements.

2.2. PHOTOGRAMMETRIC WORKSTATIONS

Conventional digital photogrammetric workstations are optimized for the handling of huge digital images. They are generally based on powerful graphic workstations equipped with special viewer components needed for the stereoscopic perception process. Moreover the manufacturers often have adopted their software solutions to the special characteristics of these workstations in order to maximize the performance.

This leads to some disadvantages. The nature of these instruments causes

- high costs
- dependence on the specialized hardware and on the manufacturer
- low flexibility to technological changes

Most of these items are caused by the proprietary characteristics of the instruments. The manufacturer has to invest great effort into an own solution which has to be amortized. The customer has to rely on the provider and his capability to keep track with the technological process. Furthermore, third party solutions being available are hardly to integrate, because of the hardware dependence of most proprietary equipment.

2.3. ALTERNATIVE

It seems promising to avoid proprietary solutions. With respect to flexibility it is necessary to have a modular and open software structure allowing to substitute individual components as there are better ones available. Furthermore one should use available standards as far as possible and insert open resources in order to reduce the development effort.

Nevertheless, the demanding characteristics of photogrammetric evaluation processes remains and has to be considered. In order to sketch just some of the needs, the following list gives an impression of the features to be met:

- simultaneous access to multiple images (only restricted by the storage capacity of the system)
- allowing images of any size and depth (up to 900 Mpixel per image, B/W and color)
- photogrammetric user environment (fixed floating mark, continuous, endless image move, multiple image and stereoscopic image viewing, registration of all kinds of object information,...)
- ergonomic environment and an agreeable, not tiring stereoscopic viewing system

These requirements need considerable computer power and an adopted control and application software. On the other hand, the technological progress on the computer market is extremely fast allowing to realize solutions based on standard PC systems.

3. DISTA

3.1. DESIGN ASPECTS

Concluding from the deficiencies of specialized solutions we use a modular concept based on available standards and corresponding resources. With respect to the display of images and handling of graphic data, for example, we use the most common standard: OpenGL. All available system platforms have at least a software solution for OpenGL and many graphic boards provide on board intelligence giving the acceleration needed for an acceptable measuring environment.

As OpenGL is supported by all common computer systems, this offers the opportunity to build a solution being independent from the manufacturer. However, the graphical user interface then has to be developed independent from the system environment too. In general, this causes some additional efforts, because UNIX and MICROSOFT™ use different graphic software. But fortunately there exist developments from other sides handling the problem of graphic interfaces in different

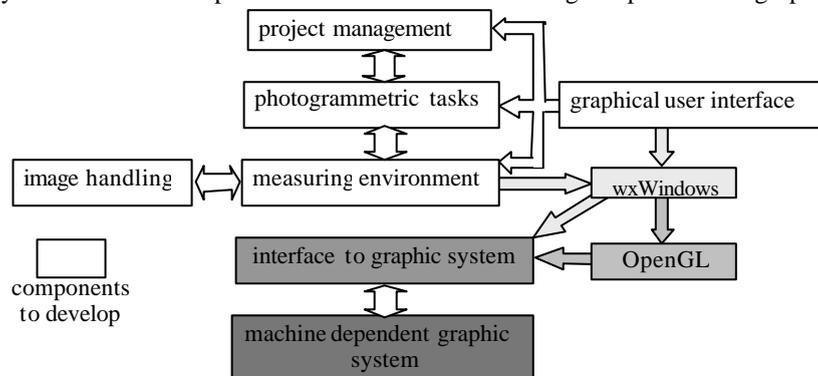


Figure 1: software components

computer environments. One very powerful solution is the wxWindows Graphical User Interface¹. It is an object oriented software package providing all necessary components to build user interfaces and is founded on system dependent libraries. This assures the look and feel the user expects and allows to install the same package on differing computer environments simply by exchanging the underlying graphic library. The user community of this package permanently grows and ranges from scientific users to ones².

The use of OpenGL and wxWindows assures the requirements coming from the wish to avoid hardware dependency, to be flexible to the technological progress and to exchange components if necessary or to incorporate new ones. Based on these standards now an application software has to be built, which has to fulfil very different functional needs (cf. Fig. 1), should reduce the effort for maintenance and modifications and should allow to compose packages being optimally structured to serve different type of applications. As a consequence, there is only one way to support all these wishes, that is to use an object oriented programming approach within C++. This language combines the advantages of object orientation with the possibility to incorporate time effective functions written in C.

3.2. PHOTOGRAMMETRY

Several tasks have to be performed when a photogrammetric project has to be handled:

- Project set up
- Interior orientation
- Exterior orientation

- Generation of stereo image pairs
- 3D data evaluation

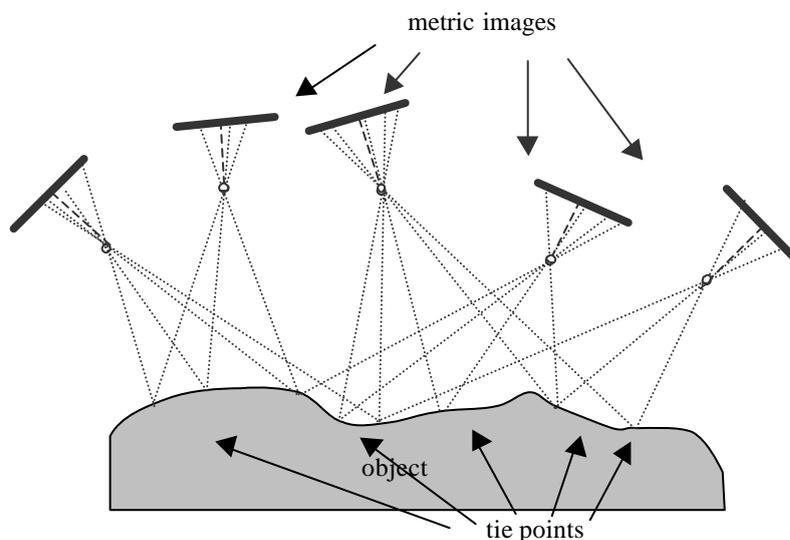


Figure 2: photogrammetric images forming a small block

This list looks very conventional for a photogrammetric processing chain but it might need much more effort than just handling two images forming a stereo model, which will be used for the data evaluation. Therefore some aspects will be mentioned documenting the flexible nature of the software developed.

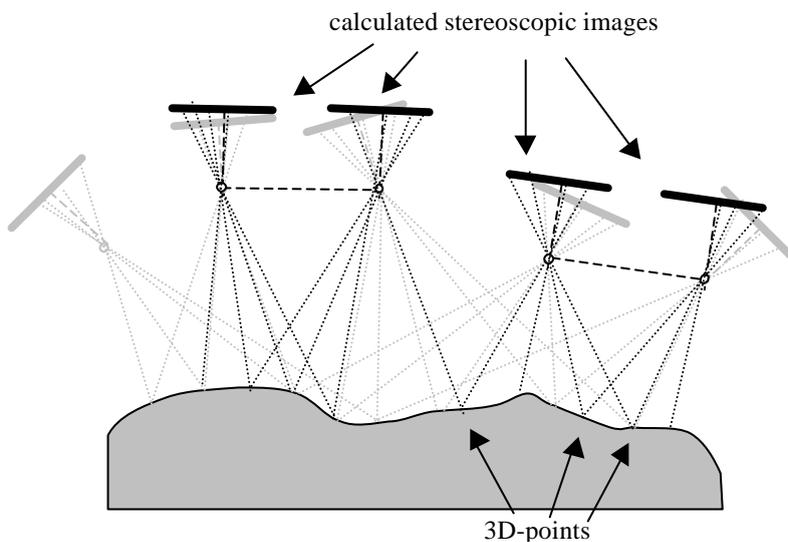


Figure 3: Generated stereoscopic image pairs used for 3D-measurements

The set up allows to store all necessary external information (point coordinates, calibration values,...) available and has to introduce the control of the following processing steps. Here flexibility is expressed by

- the number of images, which might be processed simultaneously. This number has to be only restricted by the available storage capacity
- the type of external information usable. Types have to be extended to the use of any kind of geodetic data, allowing to process distances, angles, geometric constraints and so on.

The step of the interior orientation comprises the measurement of the fiducial marks and the calculation of the geometric relation between camera system and digital system. Here additional performance is introduced, when the system allows any kind of camera type. This makes it necessary to process reseau camera data or image geometries differing from the normal central perspective model.

The step of the exterior orientation is very simple, if there exist simply one image pair. But this is not necessarily the standard situation, when surveying objects of any type. Conventionally, objects are observed by several images capturing all the parts having to be geometrically evaluated (cf. Fig.2). It's then the job of this orientation step to combine all images into a stable and fixed image block. The only way to do this is the measurement of tie and pass points expressing image rays which have to intersect and to hit the object at the same position. In principle, this might not appear as a demanding step, but if one considers, that

- images have different perspectives
- images might be exposed under differing illuminations
- most object points are signalized
- objects show occlusions and discontinuities

the process of measuring tie and pass points obviously is difficult and possibly needs some iterations until good points are found.

The generation of stereo image pairs is indispensable, because except of standard aerial image blocks there don't exist images forming a model allowing to perceive the data stereoscopically. One therefore has to introduce a processing step, which generates epipolar image pairs serving as visual basis for the 3D-evaluation procedure (cf. Fig. 3).

Finally there is the 3D-evaluation process. Here, challenges come from the fact, that the measuring environment has to fulfil the needs of a photogrammetric workplace. For example, photogrammetric operators expect to have certain control elements allowing to move through the 3D-space. They await to find the floating mark pointing to the object details resting fixed in the middle of the viewer. This makes it necessary to permanently move through the image space, what means to shift the image data continuously through the viewer – a challenging aspect, having the amount of image data in mind, which has to be handled.

The amount of image data in general introduces hard challenges. The storage amount of aerial images starts with moderate 21 MB (50 µm Pixel, B/W image) and ends with 1587 MB (10 µm Pixel, color image), when the image is prepared for an interactive evaluation, needing to provide all necessary information details. This data has to be processed during the 3D-evaluation without any narrowing of the movement during the measuring procedure. The step of continuous measuring therefore need an intelligent solution for the image display.

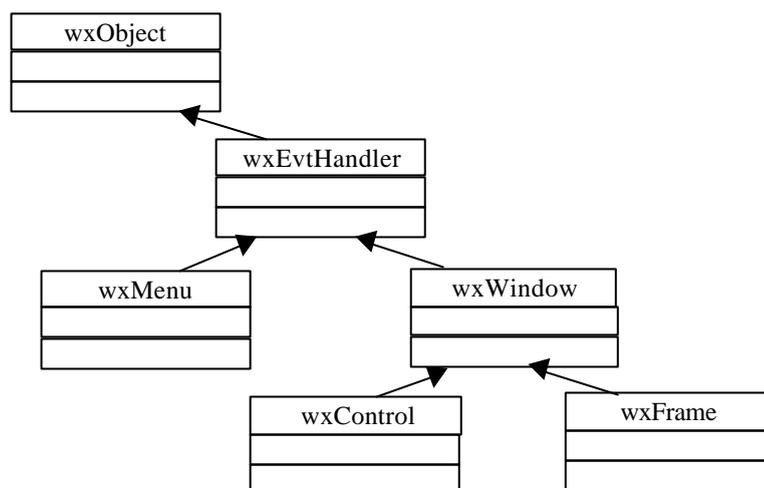


Figure 4: Simple UML showing some wxWindows classes

4. SOLUTION ASPECTS

4.1. WXWINDOWS

This package has all GUI classes needed to build a modern ergonomic user interface. For example classes for frames, controls, toolbars and menus are included. But wxWindows does not only provide GUI classes. Various helpful classes are available for handling configuration files, display a OpenGL canvas and printing preview to name only a few. To save a program configuration the native methods are used. That is the registry under Windows and plain text files under Unix. These capabilities are useful for handling project files too.

The wxWindows Framework has an excellent support by mailing lists where you have direct access to the wxWindows developer team and share your problems with other users. Questions are usually answered within the day you mailed them to the list. The whole source code base is handled via GNU Concurrent Versions System (CVS) and readily accessible for everybody. If you're interested in submitting corrections or additional features you may ask for write access.

4.2. HANDLING OF LARGE DATA SETS

4.2.1. MODULAR STRUCTURE

The software uses a modular structure for handling and displaying large data sets. It consists of an intelligent control of the image data waiting to be displayed and the display mechanism itself. This structure is applied to any kind of large images and therefore allows to view monoscopic and stereoscopic images. It is built from two major sub-components and some minor sub-components for the windowing system. Each component and their co-operation will be described in more detail in the following subsections. See Figure 5 for an UML-diagram of all classes making up this display and measure component.

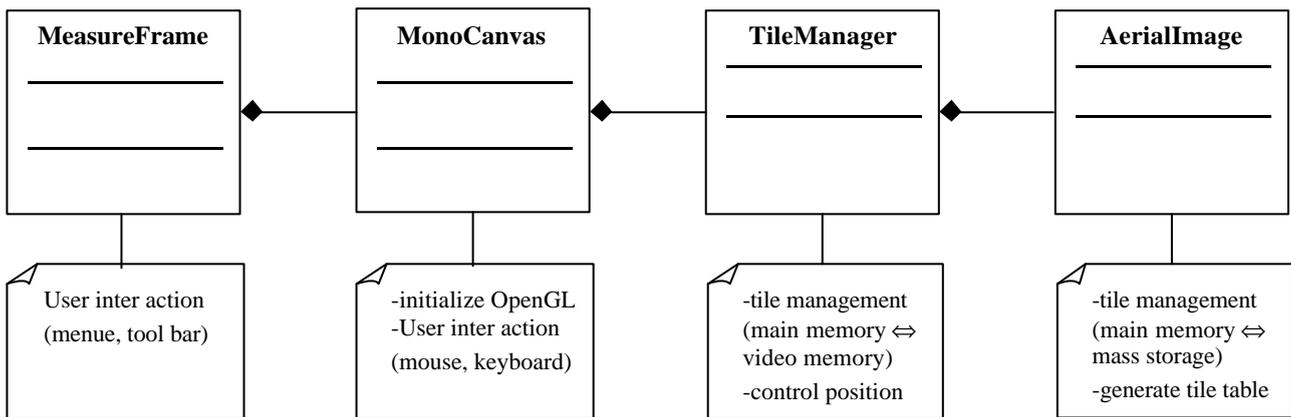


Figure 5: UML of display concept

4.2.2. IMAGE LOADING

First, a decision had to be found, how the data can be accessed with maximum speed. Several other photogrammetric packages use proprietary formats designed to easily get access to the image data. This doesn't seem to be necessary, because there exist an efficient format with tools available for data transfer: TIFF. We use TIFF, as it allows to hold all formats one can imagine. It's even possible to create an own format. In its latest version, TIFF does already provide everything needed to implement the loading component. Furthermore, TIFF is supported by most conventional image processing tools, what simplifies the data conversion from or to other sources and avoids the need to develop own conversion tools. TIFF is even flexible enough to load the images in tiles, what is inevitable with the large images sizes to be handled. With tiles one is free to load as much of the image as necessary but not more thus reducing the memory load of the application. Sam Leffler's libtiff library⁷ already comes with a small tool to tile existing TIFF images. This step is sometimes necessary in order to convert external TIFF data into tiled one, because this is not necessarily supported by all image processing tools.

With respect to the size of the tiles experience has shown, that 256 x 256 pixels for each tile are an adequate compromise between storage load and transfer speed. Other sizes are possible, but up to now there is no need to select a different one. Within the measure component the one class (AerialImage) is responsible to handle these tiled TIFF images. Fortunately the libtiff library already offers all functionality to load single tiles from a tiled image so one just has to map the corresponding libtiff functions to methods of class AerialImage. Additionally, the AerialImage class holds a tile table to handle the memory addresses of already loaded tiles.

4.2.3. IMAGE DISPLAY

Once the image file is tiled it's possible to load just those tiles needed to fill the viewport within the display window. (cf. Fig. 6). The TileManager class is responsible to manage the tiles in memory. After creation of the display window and the OpenGL viewport within this window the TileManager class calculates the number of tiles necessary to fill the viewport. It will recalculate the number every time the user resizes the display window. Next, it selects the tiles to be displayed and loads them. Then, this object waits for user interaction. When the image is moved by the operator this will be registered by the TileManager class. Dependent on the actual viewer position, it decides whether it's necessary to load new tiles and to

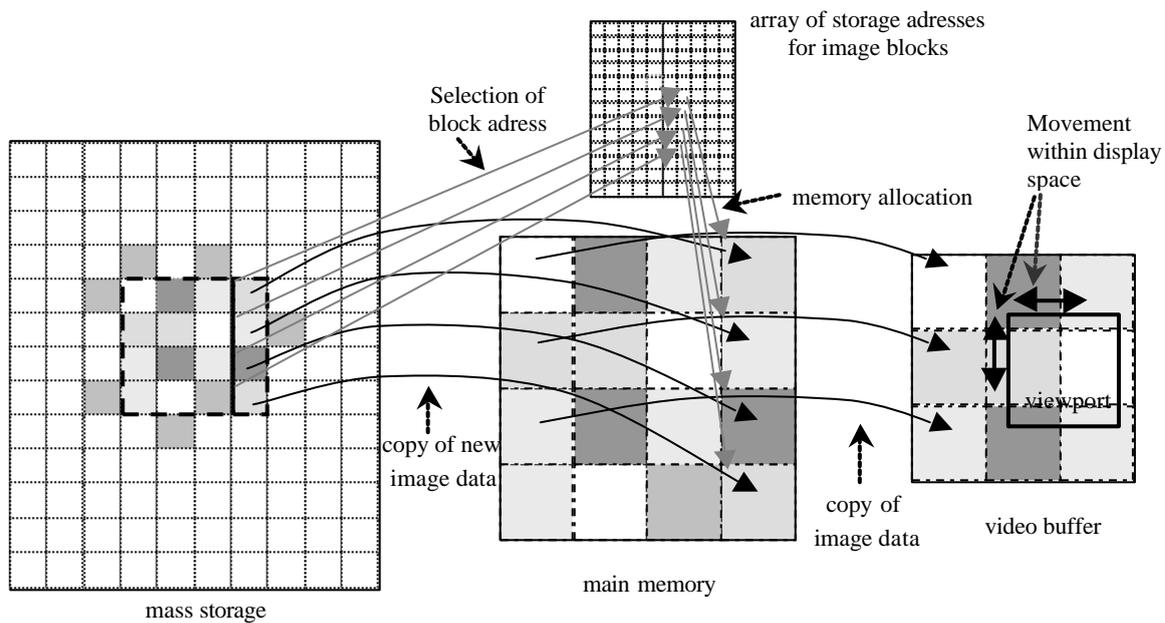


Figure 6: dynamic handling of image data

destroy old tiles or if the already displayed area of tiles has still enough data to fill the viewport. All image movement directions are handled separately. That means positive X, negative X, positive Y and negative Y movement is handled separately. As a result, the steps necessary to move the image just have to be done for one direction. For example, a movement in the negative X direction has the following processing steps:

- Calculate the current viewport offset for the tile area
- Check, if there is at least one tile width of image data outside the viewport (left side) available in the image buffer
- If less than one tile width of image data is available outside the viewport then unload a column of tiles on the right side and load a new column of tiles on the left side.

This buffering procedure makes it sure that the main memory always holds enough data, when further movement in the negative X direction occurs and new image information has to be loaded into the viewport. Obviously, there are only a few instructions necessary to handle image movement. To actually display the tiles within the viewport there're just two loops over the tile area needed (cf. Figure 6), one to display the rows and one to display the columns. OpenGL can handle raw pixel data directly so there is no need for further calculations or conversation. Experience has shown, that drawing raw pixel

data works better than using textures. These two concepts, few instructions for image movement together with fast raw pixel display using OpenGL results in very high display speed even on standard PC hardware.

4.2.4. STEREOSCOPIC VIEW

The OpenGL standard has built-in support for stereoscopic view of rendered content. Using this build-in functionality it's quite easy to get a stereoscopic view of stereo images. As all functionality to display images is encapsulated in the TileManager class it's relative straightforward to extend this class to handle stereo images. There're four things one has to take care of, when rendering stereo images.

- a stereo hardware environment is needed. That means a graphics card that supports stereoscopic view and some equipment to separate the left image for the left eye and the right image for the right eye (We use shutter glasses for image separation)
- a stereoscopic window is needed. Some operating systems and graphic card drivers support the stereo-in-window technique, where the stereoscopic view is restricted to just one window and not the whole screen (We use this technique under Windows NT)
- one has to render the content twice. Once for the left image and once for the right image (The TileManager class of the measure component handles this)
- one has to apply an offset between the left and the right image to actually create the stereo effect (The TileManager class of the measure component handles this)

So, the modular structure of the software supported by the graphics standard selected allows easily to extend the whole image management to stereoscopic images and therefore assures an effective handling of large data sets even in stereo mode and provides the functionality needed for a powerful measurement environment.

5. CONCLUSION

A modular conception based on graphic standards has been presented allowing to handle digital photogrammetric image blocks, to produce stereoscopic image pairs and to compile them performing 3D-measurements in the virtual 3D space spanned by these models. The use of the OpenGL standard even permits to use hardware acceleration providing high processing speed. Furthermore, standardization allows to incorporate from third party most of the graphic functionality necessary for user dialogs and image viewing processes, considerably reducing the effort of development. An intelligent image handling drops the amount of data transfer to the inevitable minimum and permits to handle very large images as they conventionally can only be handled within specific photogrammetric workstations.

This shows, that the use of standards together with a modular object oriented approach provides a performance up to now being reserved to high end photogrammetric workstations.

6. REFERENCES

1. J. Smart, *wxWindows Documentation*, 1999.
2. <http://www.wxwindows.org>
3. Mark J. Kilgard, *Avoiding 16 Common OpenGL Pitfalls*, 1999.
4. T. Davis, J. Neider, M. Woo, *OpenGL Programming Guide*, Addison-Wesley, New York, 1993.
5. Boochs F., Gehrhoff A., Neifer M. *An object-oriented stereo system for 3D-measurements*, ISPRS, Vol. 33, No.5, 2000
6. Boochs F.: *Photogrammetry, Script of corresponding lecture*, 1998
7. <http://www.libtiff.org>