

ORTHOWARE: SOFTWARE TOOL FOR IMAGE BASED ARCHITECTURAL PHOTOGRAMMETRY

A. Martos ^a, S. Navarro ^b, J.L. Lerma ^b, S. Rodríguez ^a, J. Rodríguez ^a

^a Metria Digital, 33428 Parque Tecnológico de Asturias, Llanera, Spain - (martos, srodriguez, jrodriguez)@metria.es

^b Universidad Politécnica de Valencia, 46022 C^o de Vera s/n, Valencia, Spain - jllerma@cgf.upv.es,
sannata@upvnet.upv.es

KEY WORDS: Orthoimage, Close Range, Photogrammetry, Architecture, Matching, Software, OpenGL

ABSTRACT:

The usual photogrammetric work-flow for producing true-orthoimages in cultural heritage depends on finding and matching homologous features among different images. This process basically relies on manually choosing a reasonable model from a set of points for a subjective proper model reconstruction of the element. This is a time-consuming, repetitive and blind recognition process requiring some spatial intuition and experience from the user. Besides, either automatic image analysis techniques (mainly image matching) or laser scanning, can be used to improve this processing. But the outputs are huge point clouds with redundant information that often is not essentially required in architectural mapping (especially when the final goal is just an orthoimage). Therefore, a combination of manual and automatic techniques seems to be the ideal tool for a production environment.

We present a novel photogrammetric software tool specially designed for production of high resolution true orthoimages of architectural buildings and sites. It depends on a simple field work consisting in a few images taken with conventional digital cameras. User interaction is simplified involving intermediate projection planes and a new raster pipeline. This reduces the need to zoom, pan, and even avoids unnecessary 3D point clouds. It is progressive and designed for non-specialized users, providing intuitive methods to visually diagnose the quality of partial results. The user interface is written in C++ using OpenGL, and all geometrical calculations are parallelized and optimized for interactive performance (10⁹ points/second), using both the central processing unit (CPU) and the specialized graphics processing unit (GPU). Improvements in image matching for posterior densification and KLT filtering for initial semi-automatic orientation were also tested. The software presented herein reduces orthoimage production time from weeks to just a few hours.

1. INTRODUCTION

1.1 Background

There are many and good academic samples successfully tested of different close-range photogrammetric recording techniques used to document architectural or archaeological sites, to produce maps, plans and sometimes orthoimages. But those usually require researchers to spend considerable amounts of time and creativity; hence very few of these techniques had scaled to commercial or serial production yet.

For unqualified users and in terms of wide production, digital photogrammetry is not yet a well established technique in cultural heritage. Theoretically it should be easy to measure from a few photographs but practical projects production still requires great amounts of computer and work time.

After years using mixed techniques to produce architectural orthoimages and plans in a commercial environment, we found that most of the limitations of photogrammetry are progressively disappearing. According to our experience, field documentation procedures are already quite optimized. It can be typically done by a small team composed of one or two persons in a single day, using lightweight equipment.

It is also possible to use just non-metric, even consumer grade digital cameras achieving very good accuracy. Field calibration allows good tolerance and there is no need to record station locations or to use stereo-rigs. For small-scale projects ground control points or topographical surveying are just optional if a reference scale is taken, using a measurement tape or a handy laser range meter.

Modern field documentation procedure is today not very different from making a detailed photographic report just following some specific guidelines.

But later processing with photogrammetric software tools still remains a bottle-neck in actual productivity. Existing software in most cases inherits the design philosophy from aerial photogrammetry with little improvements. It is focused on qualified users and it is slow and has little interactivity.

Here we show the results of the development of a software tool prototype specifically designed from scratch for heritage documentation of architectural elements. We describe the proposed work-flow and justify the computer technology used. It is intended to be easy to use for non-photogrammetrists, and as automatic as possible. It is designed to be suitable for low-budget projects, allowing easy start of quick projects with just the essential requirements and progress seamlessly towards highly accurate and detailed projects. The whole user interface design rests on a hardware-accelerated graphic pipeline specifically developed for this task.

1.2 True Orthoimages

There is no doubt that presently photogrammetry and laser scanner techniques can produce accurate and detailed 3D measurements and models. But both are just acquisition techniques; direct data often result too "raw" and has to be processed later. Additionally a great amount of cultural heritage documentation projects are small and low-budget. Additionally potential contractors and users are still not much used to handle

3D documents but they are very familiar with plans and maps that they often demand as final products.

Orthoimages are a nice compromise between plans and 3D models, which can make this transition more natural, and have some of the best advantages from both sides. Orthoimages are colourful, accurate, detailed and very objective. Are also easy to show, understand and handle, and can overlay with maps and plans. Often it even results easier to produce a high resolution orthoimage first, and then drawing the plan by copying over it.

True-orthophotographs are digital images where every point of the object is orthogonally projected with a strict pixel-level accuracy. This is different from single plane ortho-rectification or other approximate techniques, where the original photos are simply corrected and projected to an idealized perfect plane that do not exactly fits the shape of the object surface.

Generation of these *true-orthophotographs* is not very usual in architecture till now due to some drawbacks of traditional methods. Sometimes they are produced by differential rectification, projecting original photographs onto a previously known three-dimensional DSM (Dense Surface Model) which could come from photogrammetry or from other sources (Tauch and Wiedemann, 2004). As a high resolution orthoimage requires a very dense model, this implies lots of work and that is why usually laser scanner is preferred (Boccardo et al, 2002). But this is a costly solution in terms of field work, and precision of the final true-orthophoto depends on the quality and detail of that intermediate DSM.

Some other photogrammetry software produces simple two-dimensional projective transformations. But just a simple rectification, or a mosaic from a small number of them, is accurate only if the object or its parts are simple or flat and shots are taken frontally enough, so that the displacement from the real projection is small comparing to the pixel size on the final representation. When using a single plane, these errors are often evident in windows, doors or other areas far from the idealized plane (Figure 1), but as real surfaces are never completely flat, smaller errors are always present everywhere.

The next logical step to improve this method and produce still approximate but better orthoimages of simple buildings (Wiedemann et al, 2003), is to idealize or “model” the rough shape of the real object by defining a simple 3D polygonal mesh using a few 3D points on the object surface. This means defining smaller additional planes for some details, like windows or doors.

That is a extended practice for simple elements, but the problem is that even the flattest real surface is not flat at all. Therefore it may be acceptable when the object is simple and resolution requirements are low, but as it increases the method becomes impractical. When projecting more than one image onto the mesh and onto the same orthogonal plane, they should look the same. But they do not fit at all at high resolution. Smaller relief differences from the mesh become quickly evident as projection errors and the pseudo-orthoimage tend to look distorted. Also it is not unique, depending on the original image used as source.



Figure 1: Simple façade. Two plane rectified images. Differences are evident at the window but smaller errors are also present everywhere.

To deal with partial occlusions the process of composing a final orthoimage from different source photographs is unavoidable and it can be an unpleasant task if different projections do not precisely match.

Additionally it is a non-trivial task for the user to choose a good polygonal “model” that is both simple, to save working time, and that reasonably fits to the real surfaces. It requires guessing and accurately marking in advance the points that later will define a good mesh. This is a blind process requiring user experience and also takes time to process.

1.3 New graphic pipeline

Traditional photogrammetric work-flow can be summarized in choosing several features (point, edges...) from one image and looking for its homologue matches on other images. Then some calculations, like image distortion correction, rotation, projection and intersection, are used to obtain 3D coordinates for each feature. This matching work is usually performed in the space of the original photographs where perspective can make the images look quite different for the user and also for automatic algorithms.

A key concept in the design of this software tool is a novel graphic interface that allows full interactivity and corrections or refinements to be done at any time providing intuitive diagnostics from partial results. This pipeline involves very intensive calculations achieved through a software development specific for GPU (Graphic Processing Unit), using *OpenGL* (Open Graphics Library).

Great effort was put on optimization using custom algorithms to handle images and matrices directly on the Graphics Processing Unit that provides huge speed-up and the capacity to process all visualization tasks in real time. Cross-correlation, orientation and camera calibration also benefit from great performance improvements and the user interface becomes very simple and efficient.

2. METHODOLOGY

This software aims to fully cover the office work stage for orthophoto production. The typical work-flow is as follows:

2.1 Image organization

Field work flexibility is very important. Despite the fact that repeated visits to the site and extra field hours can become expensive, it is often hard to plan and organize field photography for large cultural heritage photogrammetric

projects. Some shots have to be taken from improvised locations due to limited access to some locations, variations in sun angle, changing illumination from cloud covering, occlusions (people, vegetation, vehicles...), etc.

We have designed a user interface allowing quick organization of images, not just at the beginning of the office work, but also early in the field being able to run on a mobile computer.

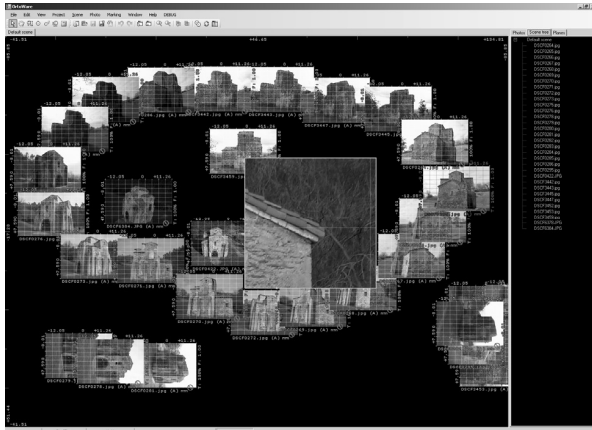


Figure 2: Desktop screen capture. Images are organized according to a spatial pattern. A magnifier permits quick inspection of details

This interface allows the user to just drag the images from disk storage or any external photo-manager software, and drop them into configurable virtual desktops. Once there, images can be interactively placed according to any criteria of choice. For instance, logical or physical organization of the project, number of common homologue marks or image quality.

To avoid memory limitations but still ensure an agile response, images that are not used for a while are automatically reduced, freeing memory for the most recently used ones. They are still promptly available in full resolution allowing quick zoom and inspection. The remaining images are loaded on demand when the user clicks on them. This was tested on an ordinary laptop with up to 400 simultaneous 16 Megapixel images with no apparent lag.

This manual organization (Figure 2) is just optional and in simple projects it can be postponed for later, if ever needed. Layouts can be multiple and as images are stored just once they can be switched at any time instantly. Later on, when external orientation parameters can be resolved, the provisional layout can naturally evolve to the 3D physical view of the model and camera locations (Figure 5).

2.2 Initial orientation

Once the project is roughly organized the next step requires the user to focus on a small set of images at a time, conveniently fitting in memory and working area, at full resolution.

Traditionally the next step would be calculating accurate relative orientation and camera positions for each image, for what a set of a few mark pairs of homologue features is needed. Afterwards bundle adjustment will calculate external matrices containing relative camera physical position and orientation for each image, allowing 3D calculations henceforth.

Early orientation is important to quickly start a new project, but the threshold of required pair marks starts from an absolute minimum of 5-8 per image, depending on the algorithm (Stewénus et al, 2006), but better 20, to handle possible mismatches and obtain a good accuracy. This requirement can consume a significant amount of time if the user has to fulfil it manually.

For proper orientation, marking has to be accurate and cover the images fully. Thus it is required either to navigate at full resolution or use a magnifier tool looking for features. This means constantly jumping between different places when there are still no helpers or constraints like epipolar lines. Often users tend to become “lazy” and avoid adding convenient or necessary new images or marks to an existing project.

This working time is reduced using a fast interface with high refresh rates. But properly choosing and marking these features is still a non-trivial task, requiring user experience and some trial and error. High contrast stickers or natural stains in flat surfaces are easy, unequivocal and accurate marks for orientation purposes, but may be of little interest to describe object geometry. Marks at edges are useful to properly model objects, but look quite different when camera convergence angle is high or if the objects are eroded. Users tend too to mark “on the air” introducing systematic bad correspondences on them, thus they should not be used for orientation.

In this software an automatic marking algorithm finds at first a few hundreds of good candidate features on each image, according to KLT feature tracker (Lucas and Kanade, 1981). Then, they are cross-correlated to find homologue matches in other images. Best relative orientation for each pair of images is determined using RANSAC algorithm (RANdom Sample Consensus, Fischler y Bolles, 1981) to discard false correspondences and determine a good set of a few homologies in short computation time (Tomasi and Kanade, 1991; Shi and Tomasi, 1994). When found these pairs are usually very accurate (1-2 pixel or better, according to our sample projects).

When a new image is first loaded, in some cases this algorithm can automatically obtain at least a dozen pairs of common features between other already oriented images. Then bundle-adjustment can establish a good initial external orientation for the new photographs with little or no user intervention at all.

If this method fails to provide the minimum of good pairs, this is easily detected in blending mode (Figure 3). Then the user might need to manually add or revise just some of them. Meanwhile, camera matrices are simply interpolated, just in case everything else fails. Right when each photograph is loaded for the first time, it is provisionally assigned with a “best guess” orientation matrix. It is obtained by interpolation from other matrices, based on the placement in the desktop working area provided by the user, i.e. if the user places the new image right between two already oriented images. This may look a too rough approximation, but the resulting matrix is often just good enough to proceed to next stages; automatic orientation or even image matching (described later).

2.3 Accuracy on orientation and calibration

Once there are enough marks, orientation is calculated “for real”, but usually still roughly. When 20 marks are available, matrices are better defined, but there is not a clear limit. More than a hundred marks produce even more reliable orientation, and even better if pairs become triplets, quadruplets, and so on.

This illustrates that all relative orientation matrices are, by nature, always provisional approximations to some degree, constructed by weighting redundant (or as seen, even insufficient) marks. Consequently those matrices can be refined at any time when more homologue marks are available.

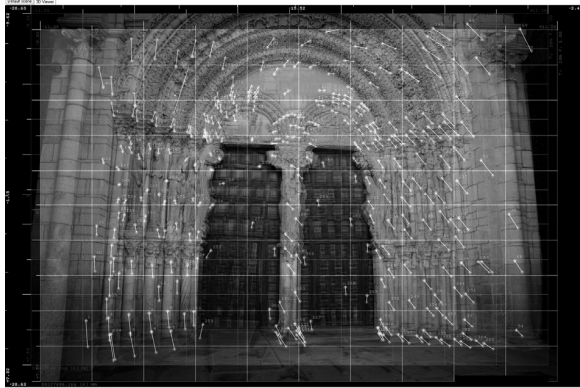


Figure 3: Portada de San Vicente. Two original images blended and overlapped for inspection. White segments join found mark pairs

Further 3D calculation such as ray intersection or image projection will greatly depend on the accuracy of these matrices, and also from camera internal model (calibration). Both orientation and calibration depend on the number of marks, their distribution over the images and object, and their accuracy. Obviously the more homologue marks, the better.

Regarding accuracy, it may look compulsory to do marking process into the original image coordinate space, as doing so on other spaces might lead to error accumulations. But as seen next, it is perfectly possible and rigorously accurate to continue marking homologues on other more convenient spaces, even without reliable orientation or calibration yet.

2.4 Graphic pipeline redesign

Our design and development depends on accelerated graphic library (*OpenGL*) to perform all point transformations in real time, including lens distortion correction and projection. For each pixel shown on the screen, many times per second and over multiple images pixels are blended at the same time. This high performance pipeline provides a few novel unique features.

Unlike in most other photogrammetry software, instead of working only into original image spaces, we can instantly switch at any time to show, navigate and mark points in other more convenient planar spaces, i.e. a rectified image space where lens distortion was removed showing straight epipolar lines, or an approximate object plane space in a fictitious surface near the object, or even the final orthophoto space.

Any update on camera orientation or internal calibration is instantly reflected in every mark feature and in every pixel shown on the screen. Although external orientation, principal point and lens distortion parameters may still be rough approximations, and therefore coordinates in new spaces will not be very accurate (Mayer, 2003), this will not pose a problem to continue working rigorously.

When new homologues are marked by the user or by any image algorithm, in any of the available working spaces, their

coordinates are seamlessly traced to original image space coordinates, and stored there. Even if the projection is not exact and if the image looks distorted the marking is surely where the user intended. These new marks will provide progressive refinements of rough camera and object model, until the desired accuracy is achieved.

The pipeline also provides a very intuitive and straightforward diagnostic method for the consistence of orientation, marking and disparity. The graphic system allows accurately overlapping and blending any set of images, changing their transparency in real time using the mouse, or switching them to front or back instantly. It is then evident at a glimpse where images match precisely and where not (Figure 4).



Figure 4: Two images blended on working plane space. Doors define the plane hence looks sharper. Parts away from the plane look blurrier

2.5 Working planes

Instead of working onto original image spaces, we found out that working onto other intermediate planes or onto the final orthoimage plane makes processing easy, robust, more accurate and faster, both for the user and for automatic algorithms.

The user starts by just roughly choosing arbitrary but convenient working planes, i.e., the main façade. Original images are projected instantly onto that plane and two or more images can be blended. For clarity, this procedure is recommended for inclinations just up to 45 degrees from the working space, choosing a perpendicular working plane for the rest, but still works fine for greater inclinations. Anyway the plane can be switched at any time.

This way the user can notice in a very intuitive way which areas fit better onto that plane, and where the images do not exactly match in blending. Some features appear displaced in different positions, becoming their disparities naturally evident. This projection can be quite good for quasi-flat surfaces, but very rough approximation for others. In a further step we can define a more detailed mesh over this surface to fit it better.

The working plane also allows the minimum user effort through defining new nodes in the mesh only where seem to be needed, and provides an immediate and clear idea of how intermediate progress adjust and what has to be added or corrected next. Matching algorithms that search for homologues can also take advantage in these spaces by using smaller search windows, becoming way simpler and faster.

2.6 Marking in blending mode

Marking new homologue points pairs is a very easy and quick task to accomplish in “*blending mode*” onto a working plane where two or more images are shown overlapped. Unlike when marking pairs into separate images, where is easy to loose the track, here the user can quickly navigate and zoom at full resolution through both images synchronized.

In the chosen sample (Figure 4 and 5) the wooden doors look already sharp, meaning that they are very close to the chosen working plane. If convenient this is the best moment to quickly mark several pairs over them, with single mouse clicks for both marks in each pair (see doors in Figure 6)

If the blended image looks a bit blurry at some area, this means it is required just a small adjustment. Instead of moving the cursor, optionally one of the images can be slightly displaced until the area looks sharp again. This action is equivalent to switch to new parallel working planes and allows marking of new points at different heights. The displacement is recorded hence just releasing the mouse button creates the complete pair.

As the cursor has to travel very small distances, it is fast and intuitive to find and mark several homologies avoiding jumps between images. Using this method marking manually hundreds or thousands of points can take just a few minutes. We experienced marking rates of up to 1000 pairs of marks in less than 15 minutes.

2.7 Mesh creation

Next step to improve the adjustment is to create a polygonal mesh that approximately models the object. In this sample (Figure 6) the right part of the object shows a simple mesh with 25 marks, created in less than one minute in blending mode. This mesh is simple, but by far enough to proceed to the next stages. The left part is intentionally unadjusted for illustration purposes. In blending both images look blurry in that area making evident for the user the need for better adjustment.

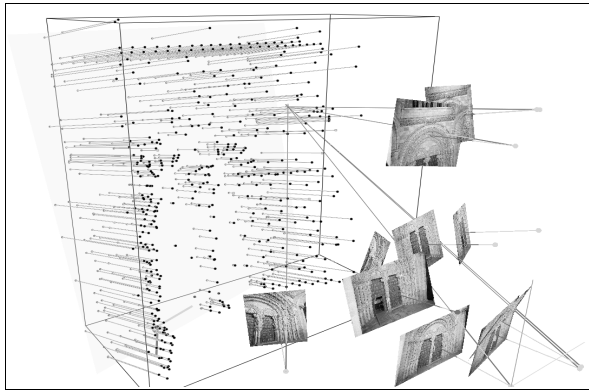


Figure 5: 3D view of the working plane used and relief errors

Instead of projecting both blended images into a well defined three-dimensional single plane, there is an alternate short-cut. Using the mesh, one image can be interactively shown “deformed” until it matches another. This has the advantage of being possible even when there is no orientation information at all and with a provisional 3D plane.

Using the deformation mesh there is not too large difference between cameras being calibrated or not, or if the features used

to orientate are not accurately located at first. It is simply enough that the mesh makes the two images to visually “match” in the areas of interest. If they do, the nodes of the mesh are new pairs, accurate and reliable as they are tracked back to original coordinates.

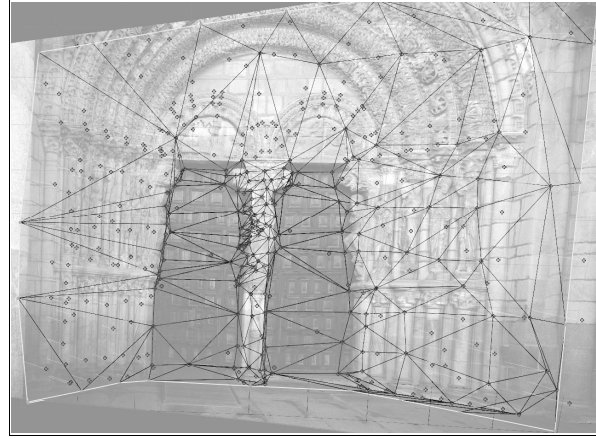


Figure 6: Mesh used. Left columns are roughly modelled by a flat mesh with no interior marks. Right side uses a mesh with just 25 points

This is a robust method, largely tolerant to errors. It provides a natural way to diagnose errors by simply watching closely areas where two or more images does not fit or look blurry. New homologies and/or new mesh nodes are created right where they are more necessary, reducing to a minimum the number of nodes.

In each step as more homologies become available they can be used to refine orientation and calibration, which in turn finally will refine projection for marking the next pairs. The user can decide to finish the process at any time depending on the immediate feedback seen on the resulting projection images in the orthoimage plane or in 3D view. When different images already match to the desired accuracy, no more work is necessary (Figure 7). Even if every mark is marked manually, the process takes the minimum working time.



Figure 7: OpenGL “near orthophoto” 3D view. Geometrical depth noise is less noticeable in this view therefore the mesh is adequate

2.8 Disparity map

For simple elements, former steps may be enough to produce good quality orthoimages and even simple but nice 3D models, but further steps are needed to reach pixel level accuracy when surfaces are complex and very detailed.

After images already roughly match in the working space, a fine detail disparity map is calculated using image correlation (Figure 8). This map provides the remaining small corrections and produces millions of new homologue points with some noise that is filtered later. Again some of these marks, the more reliable of them according to image structure, level of correlation and epipolar constraints can be used later to refine and update orientation and calibration.

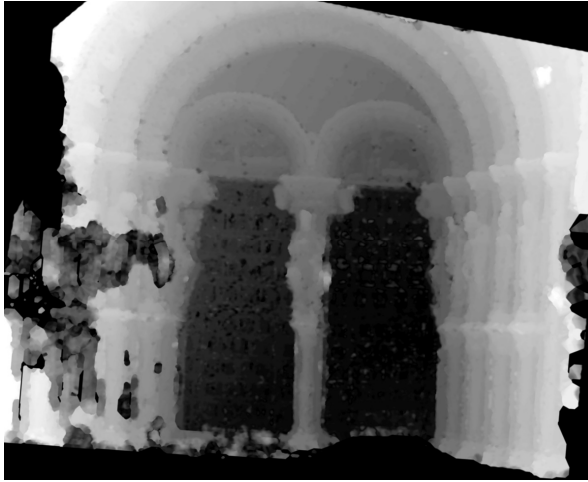


Figure 8: Fine detail disparity map (without filtering noise). Left part still requires just a few more manual marks to improve the mesh

The algorithm developed is a variation of the pyramidal correlation approach, with small search windows, optimized for parallel processing with a filling rate of 2-3 seconds per million points to provide immediate feedback and unconstrained to deal with projection errors. A sample of this map is shown unfiltered in Figure 9, where the reference working plane corresponds to the doors. Dark means low disparity, i.e. the area is more or less near the working plane, and bright means that is far from it. This information is closely related to relief as appreciated in the image hence it is easy to detect failure areas.

Image B can be corrected by this map and projected into other image space A producing a synthetic image SBA (Figure 9). Overlapping images A and SBA the matching is evident. Switching them is an immediate visual test of the quality of the map as errors are easily noticeable as residual displacements.



Figure 9: Synthetic image SBA from disparity map. Overlaps almost exactly with image A, but image information comes from image B

2.9 Orthoimage space and 3D model

After filtering and adjusting, the two original images are finally re-projected onto the real orthophoto plane (or 3D model), transparently computing intermediate corrections on working plane, mesh and fine detail disparity. Eventually, the two imagery match together providing a validated portion of the final orthophoto.

Figure 10 shows a coloured 3D point cloud, similar to those obtained from laser scanner, but just from the disparity map of the two images. It is shown unfiltered and unconstrained just for demonstration. Geometrical noise is evident in some areas and the cloud is sparse. Of course, filtering the cloud greatly reduces some of this noise and it is also possible to mesh the cloud and form a dense surface model (DSM) and even join multiple portions to increase coverage. These are well known issues in laser-scanning, out of the scope of this paper.

Traditionally each point in the images would be projected onto the DSM and then projected again onto the chosen orthogonal plane. The quality of these calculations depends critically on good camera calibration and orientation, accuracy in marking and good reconstruction of the 3D points and also a correct definition of the orthogonal plane, that in turn usually depends on everything else. Also a 3D model can be seen from any angle, therefore the mesh has also to deal with occlusions and coverage. Joining portions require seamlessly mixing the “textures” and usually reducing the model back to a reasonable amount of polygons. This is a complex process consuming memory, processing and user time and requires specialization.

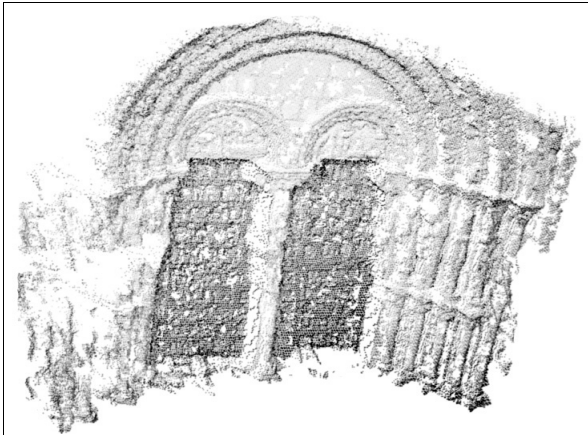


Figure 10: Unfiltered and unconstrained 3D disparity point cloud

But if the goal is just to produce true orthoimage instead 3D, production of that error prone detailed intermediate DSM is somewhat unnecessary. We can again take a short-cut and directly project the images right onto the orthogonal plane as soon as it is first defined by a few marks. This projection is not planar, it requires some 3D corrections that can be calculated right from disparities by means of a GLSL “shader”.

Again switching from different source images in blending mode, if they already match means that the deformation model fits with the orthogonal plane re-projection and thus any of the projected images is already a perfect orthoimage in that area. On blurry areas because on the relief, the 2D mesh can be used again to interactively distort one image into another until fitting. This can be converted to a soft disparity map used as starting point for next correlation levels, improving matching.

2.10 Final image composition

To produce final high-quality orthophotos for printing, former operations must be repeated using different pairs of images to deal with coverage, occlusions or shading problems. Each of this partial orthoimages constitutes a different image layer to be exported to image processing software reducing to 2D image processing from there on.

Those layers partially overlap and match precisely onto the orthoimage plane, thus the task reduces to trim evidently wrong areas, revealing the next layer, and perform radiometric equalization. If there is more than one image covering the same area (and this is usual in a photogrammetric record) it does not matter metrically which one is used as source for the final orthoimage rasterization. The choice can be done according to visual image quality factors or based on additional layers such as depth, disparity or image structure. It is planned to use texture blending from viewing angle and image resolution in model space (Petsa et al 2007).

Regardless of the working spaces, each layer is rasterized oversampled onto the orthophoto plane just once, at the desired resolution, using the same raster pipeline and directly back from the original images with no intermediate rasterizations. If original images on disk are processed in any way (re-sampled, colour adjusted, etc.) a new rasterization takes just a few seconds to be reload.

3. SOFTWARE AND HARDWARE

3.1 Graphic pipeline design

Orthoimages suitable for large-format printing are digital images requiring a resolution of 150-250 pixel/inch (Figure 11). For the software to be interactive, this typically means managing 2-4 overlapped semi-transparent several mega-pixel images, that have to be real-time refreshed on the screen. Every single pixel has to be corrected from lens distortion and perspective, and displaced to match its homologue on the other photographs.

For a good design it is critical that the whole image transformation is calculated in less than a fraction of second and repeating for more than one image at the same time.



Figure 11: Printing quality final orthoimage, seamlessly composed from five different layers after radiometric equalization

Being performance so important, all photogrammetric and geometrical algorithms have been first prototyped, tested and rewritten in C++ with performance in mind, using a combination of central processing unit (CPU) and the specialized graphics processing unit (GPU).

Drawing two overlapped mega-pixel images on the screen with all the required processing steps from original image to true orthoimage projection requires, for every single pixel:

1. Lens distortion elimination (non-linear).
2. Projective transformation.
3. Mesh intersection.
4. Fine disparity corrections.
5. Projection onto final orthoimage plane.

Drawing so high resolution images is not an easy task even if they were just directly displayed, but it is even harder if this kind of intermediate 3D processing is needed. Graphics card acceleration is obtained using OpenGL and GLSL.

OpenGL (Open Graphics Library) is a standard multi-platform library for 2D and 3D graphics developed by Silicon Graphics. The library offers the ability of GPU (Graphic Processing Unit) programming with the GLSL language (OpenGL Shading Language).

For critical tasks, our choice was to use the GPU (Graphic Processing Unit) located in the modern graphic cards. They can process several pixels (16 to 180) in parallel at the same time. It is also a 4D vector processing unit, making many vector operations naturally in one "clock step".

3.2 Interactivity

In general, graphics applications work in an iterative loading-processing-loading procedure. The images need to be first loaded in system memory and moved to video memory to be shown on the screen using the graphics library with hardware acceleration. The movement is slow and duplicates memory usage. Then calculations are performed in CPU in a pixel by pixel sequence. This process is very slow because the CPU is not optimized for pixel (image) calculations. Finally, the resulting image data has to be moved again to video memory using *Windows GDI* (Graphics Device Library) so the user can see the resulting transformation.

The solution for efficient real-time graphics transformations is programmable hardware GPU. This new approach offers the ability to process all the calculations in the graphics card, accessing the video memory directly. Furthermore, the GPU is designed specifically to work with images and algebraic calculations, so many of the vector and matrix operations become native, and it also has a concurrent and parallelized design which provides the ability to perform the operations in many pixels at the same time. With this GPU approach it is possible to avoid the load-process-load stages, because only one initial load is needed.

Combined with the specific design of the GPU for working with images and algebraic calculations, our initial tests showed speed-ups of 1000x in processing time on consumer grade (100€) graphics card (*NVidia Geforce 8500*).

Another fact for using a GPU system for the project, is the rapid evolution that this kind of hardware is experiencing, provided by the video-game industry. Due to it, GPU are evolving more quickly than CPU processing.

Other more complex algorithms like disparity calculations are still being tested in CPU before being ported to GPU. To keep interactivity the application runs this kind of calculations in background avoiding to stall the program interface (the hourglass cursor that other programs show when they are processing) thus the user can continue working. When it finishes, results are shown immediately on the screen.

4. CONCLUSIONS

This paper shows some early results on high interactive true ortho-imagery reducing user intervention to a minimum. *Orthoware* is a very easy to use computer program capable of fully processing a total of several billion (10^9) points per second in a conventional computer system. This is just departing from photographs taken with ordinary digital cameras in affordable field work. Office production time may reduce to a few hours (or even minutes) instead of weeks. Accuracy is self-evident in

form of blurry areas that are easy to correct just by making small adjustments until they become clearer.

Orthoware comes out after several years of software development, bearing in mind commercial production. The novel characteristics of its interface are only possible thanks to the standardization of recent affordable high performance graphics systems for video-games industry.

At the time of writing this paper, the software and the methodology are still in development, being tested in parallel with data coming from current production projects that are being executed. This development will simplify the overall production of high quality orthoimages in the field of architecture, opening new ways for non-specialized users and helping to successfully consolidate ortho-imagery in cultural heritage documentation.

5. REFERENCES

- Boccardo, P., Dequal, S., Lingua, A., Rinaudo, F., 2002. True digital orthophoto for architectural and archaeological applications. GISdevelopment.net. <http://www.gisdevelopment.net/Application/archaeology/general/archg0002.htm> (accessed 28.Sep. 2007).
- Lucas, B.D. Kanade, T., 1981. An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pp. 674-679.
- Mayer, H., 2003. Robust Orientation, Calibration, and Disparity Estimation of Image Triplets. *Lecture Notes in Computer Science*, Vol. 0302, pp. 281-288.
- Petsa E., Grammatikopoulos L., Kalisperakis I., Karras G., Pagounis V., 2007. Laser Scanning and Automatic Multi-Image Texturing of Surface Projections. *XXI International CIPA Symposium*, 01-06 October, Athens, Greece.
- Shi, J., Tomasi, C., 1994. Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600.
- Stewénius, H., Engels, C., Nistér, D., 2006. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4), pp. 284-294.
- Tauch, R., Wiedemann, A., 2004. Generation of Digital Surface Models for Architectural Applications with Archimedes3d. *IAPRS*, Vol.XXXV, Part B5, pp. 350-353.
- Tomasi, C., Kanade, T., 1991. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April.
- Wiedemann, A., Moré, J., Tauch, R., 2003. ARCHIMEDES3D- An Integrated System for the Generation of Architectural Orthoimages. *International Archives of Photogrammetry and Remote Sensing*, XXXVIII, pp. 554-558.