IMAGE BASED ARCHITECTURAL TRUE-ORTHOPHOTOGRAPHS

A. Martos [a], S. Navarro [b], J.L. Lerma [b], S. Rodríguez [a], J. Rodríguez [a], J. González [a], F. Jordá [a], M. Ramos [a], A. Pérez [a]

[a] Metria Digital, 33428 Parque Tecnológico de Asturias, Llanera, Spain, -
(martos, srodriguez, jrodriguez, jorgegl, fjorda, mramos, aperez)@metria.es
[b] Universidad Politécnica de Valencia, 46022 Cº de Vera s/n, Valencia, Spain, -
jllerma@cgf.upv.es, sannata@topo.upv.es

KEY WORDS: Orthoimage, Close Range, Photogrammetry, Architecture, Correlation, Calibration, Programming.

ABSTRACT:

True-orthophotographs are now being widely used in some areas, proving already as very valuable documents even replacing maps being more complete, reliable and objective. But they are still hard to produce with enough quality for other uses and scales, such as in architecture and conservation. Most works in these fields are just approximate ortho-rectifications by single or multiple planes [1]. Some academic attempts to reach quality "true" orthophotos were made recently [2] [9] using a combination of laser scanner and images, but are still costly solutions lacking from the flexibility needed for field work.
In this paper, a new practical technique and software to produce these architectural orthophotographs is presented, departing just from conventional digital photographs with no need for metric cameras or laser scanner, and for small projects, not even for topographical surveying. It is also intended to be easy to use for non-photogrammetrists, dramatically reducing the working time usually spent in other photogrammetry applications.
This method requires very intensive calculations through a dedicated software development using GPU (Graphic Processing Unit), making easier and faster the production of high quality true-orthophotographs. The prototype was already tested producing some orthophotographs in a commercial environment and is being now developed as a compact software application making very easy to widely produce quality architectural orthoimages, that we feel can become a standard in heritage documentation soon.

1. INTRODUCTION

We refer to true-orthophotographs as digital images where every point of the object is strictly orthogonally projected with 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 a idealized perfect plane that do not exactly fit the shape of the object surface.

Generation of these *true-orthophotographs* in architecture is not very usual, due to the difficulties of the traditional methods. Traditionally quality orthoimages are produced by differential rectification, projecting original photographs onto a previously known three-dimensional DSM (Dense Surface Model) that could come from photogrammetry or other sources. As a high resolution orthoimage requires a very dense model, usually laser scanner is preferred. But this is a costly solution in terms of field work and precision on the final true-orthophoto depends on the quality and detail of the DSM.

Also usual in cultural heritage documentation is three-dimensional plotting. That traditionally depends on the realization of stereoscopic shots maintaining a convergence angle not far from 5º and external orientation processing that needs the knowledge of a few GCP (Ground Control Points) generally measured by topographic methods involving the use of metric or semi-metric cameras. More recently these techniques started to use monoscopic views, field calibration, and non-metric cameras, giving much more flexibility to the field work. But office work still remains mostly manual, slow and repetitive, being very time consuming for the user.

Photogrammetry software normally implements two-dimensional projective transformations. But a simple rectification, or a mosaic of them, is accurate only if the object or its parts are flat, and shots are frontal enough so 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 more evident in windows, doors or other areas far from the idealized plane, but as real surfaces are never flat, smaller errors are always present everywhere (see Figure 1).

Figure 1. A pair of single plane rectified images.

A usual practice for producing low resolution approximate orthoimages [3] [15] is to idealize or "model" the rough shape of the real object defining many plane surfaces, one for each different part. Choosing a simple polygonal "model" that fits to the real surfaces is a non-trivial task, requiring some user experience and manual work. This is true also for more complex primitives.

Even the flattest real surface is not flat at all. Therefore, when resolution increases, differences can become quickly evident when projecting more than one image onto the same plane. The process of composing a final orthoimage from different source photographs (unavoidable due to occlusions) can be an unpleasant task if different projections do not precisely match between them (see Figure 2).

Figure 2. Two projections overlapped on the same single plane. Left window is closer to approximate plane, so looks sharper

A new practical technique and software to produce these architectural orthophotographs is presented, departing just from conventional digital photographs with easy field work. There is neither need for metric cameras nor laser scanner and, for small projects, not even GCPs or topographical surveying. It is also intended to be easy to use for non-photogrammetrists, and as automatic as possible. It is designed to be suitable for low-budget projects, being easy to start a quick project with just the essential requirements. Furthermore, it allows to progress towards highly accurated and more complex projects by working after with detail in smaller parts.

It is a monoscopic process where photos can be taken with any camera and with almost any convergence angles (if frontal enough). The only equipment required would be an ordinary digital camera and some other lightweight and affordable tools such as stickers, a tape measure and a plumb-line for scale and levelling data, respectively, reducing field work considerably.

To achieve these objectives, it was needed to redesign the usual methodology used by other photogrammetric software, following a set of key directive requirements:

1. All procedures automatic or semi-automatic.
2. Easy to use for non-specialized users that just have to inspect partial results and make trivial corrections.
3. High interactivity, displaying changes fast. All calculations should be real-time or almost real-time.
4. Progressive procedure, produce rough results quickly but allowing later refinements adding more data.
5. Accurate and detailed results, providing visual methods to visually diagnose partial results quality.
6. Affordable and flexible. Field work requires just light affordable equipment and software should run in an ordinary computer or even a laptop.

The proposed method is based on changing interactively the user workspace from the original images to the more natural final ortho-image approximate plane (OA). Coordinates are translated on the fly and always referred to original images to keep consistency and allow refinements at any time.

Obtaining homologies between millions of points in this space is an easier task and producing detailed disparity maps is fast. This does not require any previous knowledge of the DSM of the object. In fact, in some cases this procedure is able to produce accurate true-orthoprojection even when the 3D structure is ill-determined due to low intersection angles. Instead, DSM becomes an indirect sub-product obtained after the orthoimage, instead of being a mandatory requirement.

2. METHODOLOGY

Ortho-projection without DSM

The usual photogrammetric workflow can be summarized in choosing a feature (point, edges, ...) from one image and finding its homologue match on other images. Then some calculations like distortion corrections, rotations, projection and intersection are used to obtain each 3D feature [16]. This is performed in the space of the original photographs where perspective can make the pair look quite different for both the user and for automatic algorithms [4] [5].

Instead, we found out that working directly onto the final orthoimage space/plane makes the processing simpler, robuster, more accurate, and faster, both for the user and for automatic algorithms. If all images are approximately projected to this plane, the user can see errors in a intuitive way, does the minimum effort defining new surfaces only were needed and has an immediate and a clearer idea of how results progress and what have to be added or corrected next. Correlation techniques that search for homologies can use smaller search windows and so simpler and faster algorithms.

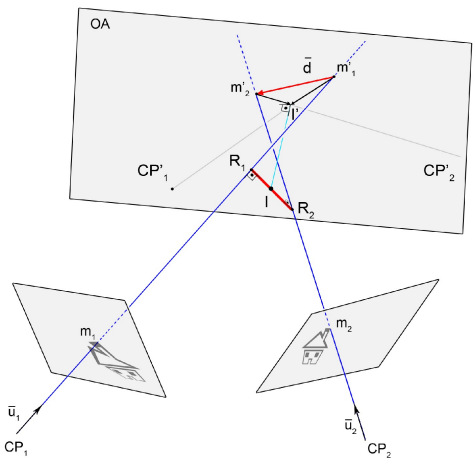

Figure 3. Intersection of rays and projection from disparity

Figure 3 shows (not at scale) the rays of two homologue marks $\mathbf{m_1}$, $\mathbf{m_2}$ from two different photographs with projection points $\mathbf{CP_1}$ and $\mathbf{CP_2}$. The real 3D position of the object is **I.** It is obtained from either 3D laser scanner data or photogrammetry software that calculate where rays intersect. Then **I** is

orthogonally projected to approximate orthophoto plane **OA**, yielding point **I'**.
But **I'** can also be calculated with great approximation (less than one pixel if its intersection is good) directly on **OA** plane just from the plane coordinates. If the homologies are calculated in plane **OA** (no matter that **OA** is not exactly the plane that fits the surface), knowing for instance **m1'** and the disparity vector **d**, one can reconstruct **I'** without precisely knowing the 3D position of **I** (for instance in low angle cases)

At first, the plane is an approximation as well as the lens distortion external orientation rough values. Despite involving millions of points, all calculations are done on the fly so every change in internal or external parameters due to calibration or fine orientation are shown instantly. Disparity calculations in this plane remain unequivocal and can still be traced back to the original photo-coordinates for further calibrations or orientations refining, if needed.

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 and look blurry. This can be corrected with new homologies right where they are more necessary. Once finished the processing, all the image texture should match.

Set-up and approximate orientation procedure

As starting point when images are loaded, to quickly establish an approximate external orientation for the cameras, a semi-automatic algorithm finds a few hundreds of good homologue features according to the KLT (Kanade-Lucas-Takeo) feature tracker and correlated, on the original photographs. Then relative orientation for each pair of images is determined using RANSAC (RANdom Sample Consensus, Fischler y Bolles, 1981) algorithm to discard false correspondences and determine a good set of a few homologies in short computation time [10] [11] [12] [13]. This orientation is checked against residuals by bundle adjustment.

In our sample project, after the relative orientation, a maximum residual error of 1.48 pixels with a standard deviation of 0.30 pixels was obtained, proving to be a good orientation result (see Figure 4).

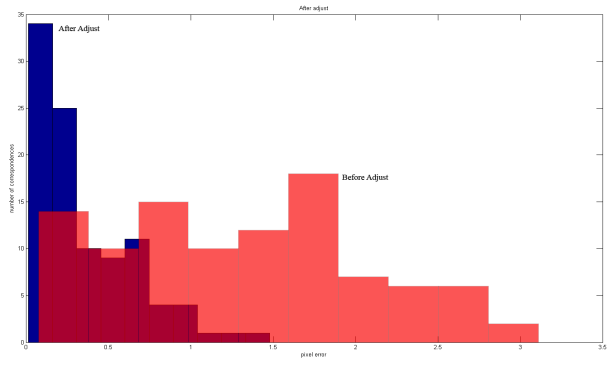

Figure 4. Mark residuals before and after bundle adjustment

In the worst case, when this automated orientation fails, the user has to correct some of the matches or mark an alternate set of 6-10 pairs (until initial orientation is achieved, Fig. 5).

At this point, external orientation, principal point and lens distortion parameters are still very rough approximations [14]. But there is not a large difference between calibrated or uncalibrated cameras, or whether the features are not accurately located. It is enough that the two images "match" at least for some areas of the object. Rough errors can be manually corrected or refined later by adding a few more reliable marked homologies.

This is a very easy and quick task to do in "blending mode" where two or more images overlap while the user can quickly navigate and zoom through them at full resolution. Manually marking hundreds orthousands of points takes little time.

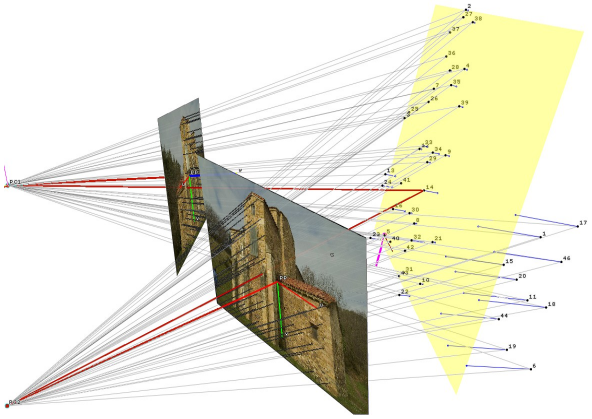

Figure 5. 3D OpenGL view of external orientation and approximate ortho-plane. Photos are lens distortion corrected

Right then, the user defines a working plane and all original photographs are processed and projected onto this ideal plane in real time. As this projection is still approximate, the images do not exactly match yet when the images are blended together. In fact, some features appear in different positions, becoming their disparities naturally evident.

Then using semi-automatic correlation techniques, a disparity map is produced and applies the needed displacement as corrections. This bring out millions of new homologue points used to instantly refine and update external and internal parameters, and even the camera model (with field recalibration) until the reaching the desired pixel accuracy.

Figure 6. Disparity map detail of the lower part of a façade. Bright areas have more disparity due to relief

A sample of this methodology is shown in Figure 6, where the approximate planes fit more or less the main façade. Columns are too far from this plane for the algorithm so are treated as a different plane and are left for another round. For the rest, dark means low disparity i.e. the area is near the plane, and bright means that is far from the plane. This information is related to relief and is appreciated in the image, but it is not the goal of the algorithm to get relief but proper ortho-rectification according to Figure 3.

Using this disparity map in both directions, one image can be corrected and projected onto another to check the accuracy in the sharpness of the blending (see Figure 7). Then both images are projected onto the real orthophoto plane so they finally match together providing a validated portion of the final orthophoto.

Figure 7. One image is mapped into another and matching is evident. Red areas have no disparity available from this pair

This operation can be repeated on different pairs of images to reveal occlusions. At the final stage, user intervention is reduced to decide which photograph is used as source for each area of the final orthophotograph, because all of them match precisely onto the orthoimage plane.

In each of these steps more homologies become available. As all calculations are still done on the fly, orientation and recalibration can be again refined at any stage as represented in Figure 8. This makes the project progressively more accurate as the work evolves.

A final true orthophotograph and a 3D model are shown in Figure 9 and Figure 10 respectively.

Software and Hardware

As reference, lines in drawings can be as narrow as 0.2mm wide. So to replace them with orthophotos, suitable for large-format printing, a resolution of 150-250 points/inch is required for making highly detailed digital images. For the software to be interactive, this typically means 2-4 overlapped semi-transparent Megapixel images, being real-time refreshed on the screen. So, every single pixel has to be corrected from distortion and perspective, and then slightly displaced to match its homologue on the other photographs.

It is critical for the whole image transformation to be calculated on the fly, in less than a fraction of second and repeated for more than one image at the same time.

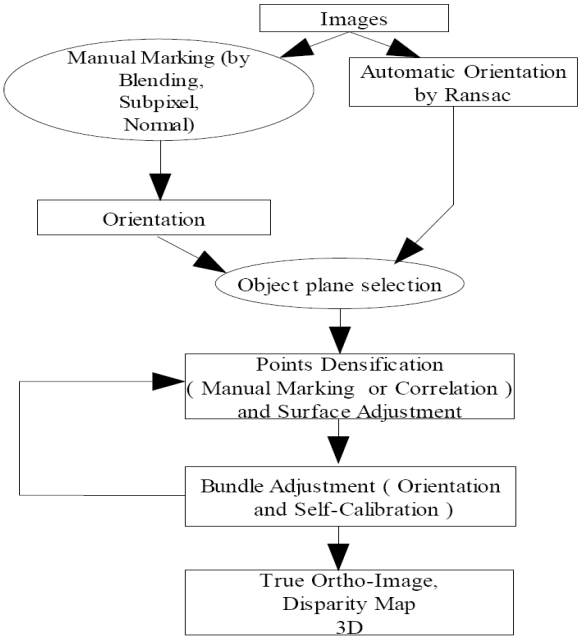

Figure 8. Work flow for calibration and orientation stages

Being performance so important, all photogrammetric and geometrical algorithms have been first prototyped and tested using Matlab, and then rewritten in C++ with performance in mind, using convenient matrix algebra, parallelized and optimized for performance, and using a mixture of central processor (CPU) and the specialized graphics processing unit (GPU) in the graphics card.

Drawing lots of high resolution original images is not an easy task even if they are directly displayed, but it is even harder if this kind of 3D processing is needed. Simple graphic card acceleration is obtained using OpenGL + GLSL (OpenGL shading language) [6].

OpenGL (Open Graphics Library) is a standard multiplatform library for 2D and 3D graphics. It was initially developed by Silicon Graphics in 1992. Since version 2.0 the library offers the ability of GPU (Graphic Processing Unit) programming with the GLSL language (OpenGL Shading Language). GLSL is a language very similar to standard C with a set of math operators that matches the native instruction set of the GPU, allowing a low-level access to the hardware, but maintaining the successful programming model of C [7] [8].

For real numbers, drawing two overlapped 12 Megapixel images in the screen with all the required processing steps from original image to true ortho image projection requires, *for every single pixel*:

1. Lens distortion elimination (non-linear).
2. Projective transformation.
3. Disparity correction.
4. Projection onto orthophoto plane.

This takes several minutes in Matlab, while the optimized C++ version, with multi-thread implementation takes just 2 seconds in a 3GHz dual core computer. But this is still not enough fast for true real-time operation. Single or even multi-core CPUs still can provide only a 2x or 4x speed improvement, not enough for a true real time refreshing.

An interesting solution is to use GPU (Graphic Processing Unit), the 'CPU' located in the modern graphic cards, that can parallel process several pixels (16 to 80) at the same time. It is too a 4D vector processing unit, making many vector operations naturally in one "clock step"

So all image processing algorithms were redesigned and rewritten, parallelized and vectorized specifically for GPU programming using GLSL (OpenGL Shading Language)

Using a consumer 100 Euro graphics card (NVidia Geforce 8500), the same calculations were finally done 240 times per second achieving the desired performance.

Design of the graphic pipeline

Although GPU is much faster than CPU, it is instead a more constrained platform. In current graphic cards, GPU memory is limited to about 512 MBytes that should be enough but can not be directly accessed. An arbitrary number of images can be stored until the memory is full, but OpenGL is limited to 2048x2048 or 4096x4096 pixels in the maximum dimensions it can handle as a single 2D texture. High quality digital photographs can easily exceed this size.

To overcome this limit, the images are internally stored and addressed using instead 3D textures. For these (usually used for medical volume visualization) the size limits are between 512x512x512 to 2048x4048x2048 pixels. This allows much more information to be directly addressed. So original 2D images are sliced and the algorithms are designed so memory access is translated from 2D data access to 3D data access. This is transparent for the user but allows a high degree of optimization for parallel processing.

With this frame-rate, several images can be processed simultaneously in a fully transparent way for the user, without any noticeable slowdown, giving the application a smooth behaviour and an interactivity far from being known until now in photogrammetric applications.

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 screen using the graphics library with hardware acceleration. This movement is slow and duplicates memory usage. Then calculations are performed in CPU in a pixel by pixel sequence. This process is also 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 are executed with native instructions, and it also has a concurrent and parallelized design which provides the ability of perform the operations concurrently in many pixels at the same time.

Figure 9. Final scaled orthoimage, composed from 9 different façade portions that match precisely

So with this GPU approach it is possible to avoid the load-process-load stages, because only one initial load is need. Combined with the specific design of the GPU for working with images and algebraic calculations, our initials tests showed speedups of 1000x in processing time, and this using mainstream graphics hardware.

Another fact for using a GPU system for the project, is the rapid evolution that this kind of hardware is obtaining, provided by the video-game industry. Thanks 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) so the user can continue working. When it finishes, the results are show immediately on screen.

3. CONCLUSIONS

This paper shows some early results on true ortho-imagery with high levels of interactivity, and an easy to use computer program already capable of fully processing a total of several billion (10^9) points each second in a standard PC. This is departing just 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.

This would make very easy to widely produce quality orthoimages in architecture. We guess that the overall ortho-imagery processing might become a standard documentation procedure soon.

Figure 10. 3D model showing the planes used to approximate the surfaces

REFERENCES

[1] D. Liebowitz, A. Zisserman, "Metric Rectification for Perspective Images of Planes," *cvpr*, p. 482, 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98), 1998

[2] Boccardo, P., Dequal, S., Lingua, A., Rinaudo, F., 2006. True digital orthophoto for architectural and archaeological applications.GISdevelopment.net. http://www.gisdevelopment.net/application/archaeology/general/archg0002pf.htm

[3] 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.

[4] Hartley, R., Zisserman A. Multiple view geometry in computer vision. Cambridge University Press

[5] Kraus, K., 1997: Photogrammetry, Volume 2, Dümmlers Verlag

References from Other Literature:

[6] Randi J. Rost, January 25, 2006, OpenGL Shading Language, Addison-Wesley Professional; 2 edition, #ISBN-10: 0321334892# ISBN-13: 978-0321334893

[7] Dave Shreiner , Mason Woo, Jackie Neider, Tom Davis, August 1, 2005, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Addison-Wesley Professional; 5 edition, #ISBN-10: 0321335732 #ISBN-13: 978-0321335739

[8] Microsoft Corporation, 2007 Microsoft Developer Network, GDI+,http://msdn2.microsoft.com/en-us/library/ms533798.aspx

[9] J.M. Biosca, S. Navarro, J.L. Lerma Modelado tridimensional de una bóveda barroca mediante Láser escáner y de fotogrametría terrestre. Los angeles músicos de la catedral de Valencia

[10] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. International Joint Conference on Artificial Intelligence, pages 674-679, 1981.

[11] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[12] Jianbo Shi and Carlo Tomasi. Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, 1994.

[13] Stan Birchfield. Derivation of Kanade-Lucas-Tomasi Tracking Equation. Unpublished, January 1997.

[14] H. Mayer. Robust Orientation, Calibration, and Disparity Estimation of Image Triplets. Institute for Photogrametry an Cartography, Bundeswehr University Munich

[15] Moons, T., 1997. Report on the Joint ISPRS Commission III/IV Workshop "3D Reconstruction and Modeling of Topographic Objects", Stuttgart, Germany. http://www.radig.informatik.tu-muenchen.de/ISPRS/WG-III4-IV2-Report.html (accessed 28 Sep. 1999)

[16] Lerma Garcia, J.L., 2002. Fotogrametría Moderna: Analítica y Digital. Universidad Politécnica de Valencia. 550 páginas.