Logistic problems at the Institut Cartogràfic de Catalunya

M. Torre, A. Magariños, J.L. Colomer

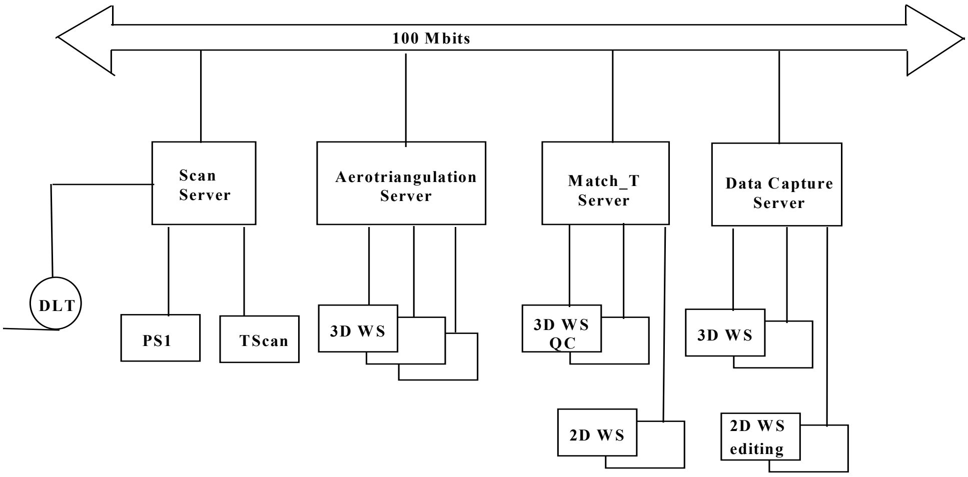
Institut Cartogràfic de Catalunya
Barcelona-Spain

1. Introduction

Our work had two goals: first, to reduce the operator interaction as much as possible to save time and lessen errors and second, to centralize data and run batch processes in servers. We believe that although the tools built are very specific to the Intergraph environment, the paper can bring some ideas for users of other systems.

2. System architecture

Nobody will argue against the benefits of keeping files on-line for the duration of a job step. For example, in point transfer we measured 4% of the operator time devoted to data management. So we decided to configurate generously our servers with as much as 100 GB to 300 GB per server as shown in the figure below.

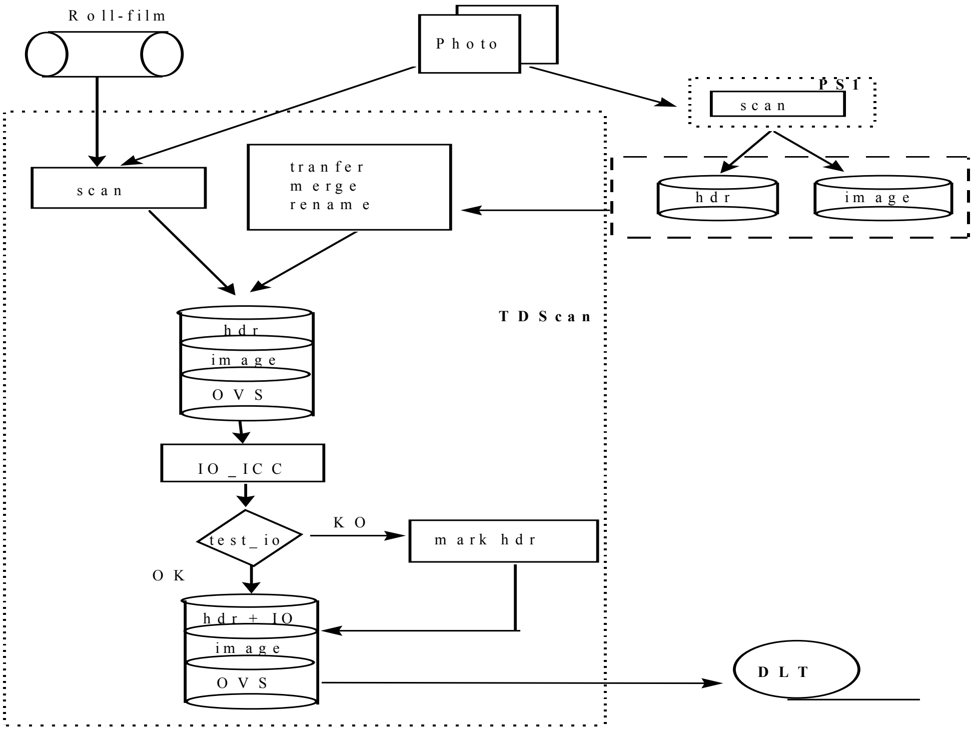


Note that we assign a server to each of the digital photogrammetric tasks: scanning, point transfer, DTM generation and stereoplotting. Batch jobs are prepared in 2D workstations and executed on the servers. Point transfer and visual checking of Match-T results are performed on workstations with files mounted remote on the servers. Stereoplotting is still performed with the images on local disks.

3. Scanning.

We operate now a PS1 and a TDSCAN with the Autowinder feature for scanning rolls of films in unattended mode. For a long time we have been waiting for the possibility of adding our own information to the header of the raster files. Besides storing standard information related to the scanner and the scanning process, we wanted to store any type of information that could optimize our workflows or could be of use for long term archival. Right now, the TDSCAN software provides for this capability

However, this is not the case for the PS1, which stores the information of the scanner and the scanning process in a separate ASCII file. In addition, the file headers can be only accessed with a piece of software that is only available on NT. Therefore we had the problem of homogenizing the header of the raster files. We solved it by transferring the image file and the associated ASCII file to the NT environment and then use the file access library (CFL) for storing the ASCII data into the header. We also wanted to extend the file name to allow better identification and automation in the workflow, so we take advantage of having the file in the NT environment for extending the 14 character UNIX file name.



We also wanted to store the measurements of the fiducial marks and the parameters of the inner orientation, and we wished to do this automatically. Although Intergraph provides a tool for doing inner orientations, we implemented our own which uses a library of cameras with templates of the fiducial marks, calibration data and a threshold for the residuals of the inner orientation. The software performs a check of the wrong/right reading of the film, locates and measures the fiducials and performs the inner orientation. If the residuals are above the thresholds, a flag is set in the header.

The result is a normalized raster file with a meaningful file name and a header containing the inner orientation. This file is stored on a DLT and catalogued in a tape library for later use.

4. Design of the photogrammetric project.

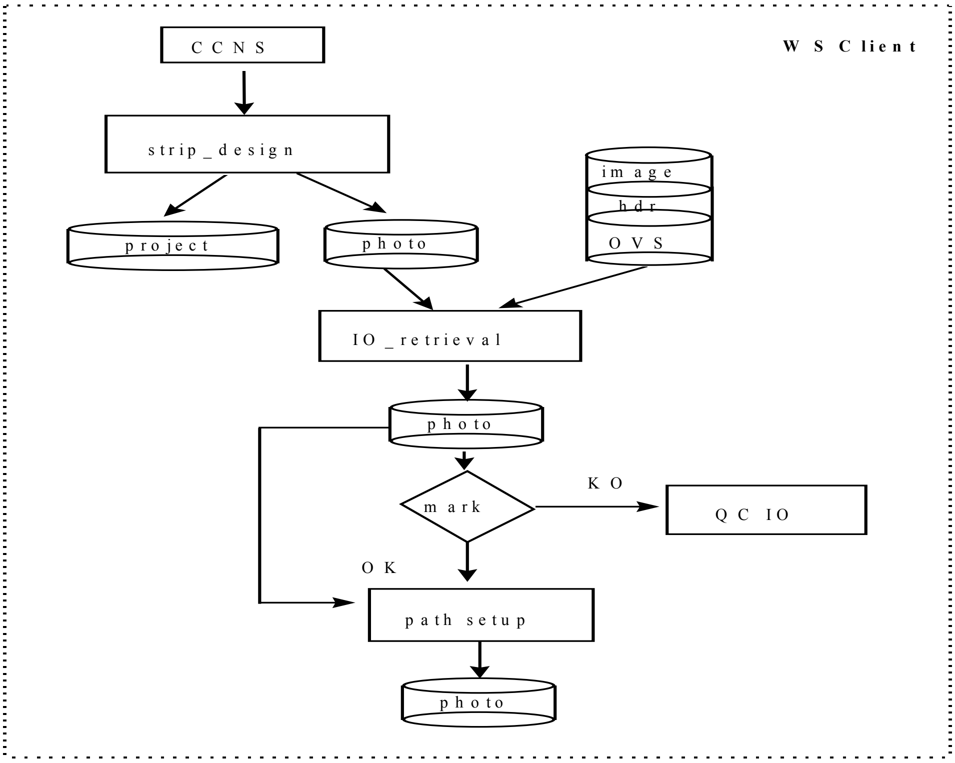
Point transfer and aerotriangulation are the next steps. As stated before, the point transfer is performed with all the image files of a photogrammetric block mounted remote on the 300 GB server. The Intergraph software needs to create two files for this step:

- a *project* file that stores general information and the parameters and defaults of the project
- a *photo* file that has an entry for each photo with the path to the image file, the footprint of the photo, the coordinates of the projection center, the image coordinates of the fiducial marks, the IO transformation parameters and the measurements of the pass and control points in image coordinates.

In the standard environment, the operator uses a GUI to define the strips and create the files. Instead, our "strip design" software automates this by:

- creating the file with one entry for each photo, ordered stripwise
- automatically creating the file path by searching for the file name in all the volumes mounted (this utility is used on relocating files in other volumes/directory and is based on AI techniques to speed up the search)
- retrieving the IO measurements and parameters from the header of the image file, checking the error flag that was set if the residuals were below tolerances, branching to interactive actions if error, and storing the IO in the *photo* file.
- reading the output from the Computer Controlled Navigation System (CCNS-IV) storing the GPS centers of the photos and computing and storing its approximate footprint. This information is used by the Intergraph software to help the operator in locating the conjugate points by driving him automatically to the overlapping areas.

The result is a *photo* file ready for the interactive point measurements task. When complete, we run the block adjustment using our GeoTex software and transfer the adjusted points back and update the *photo* file.



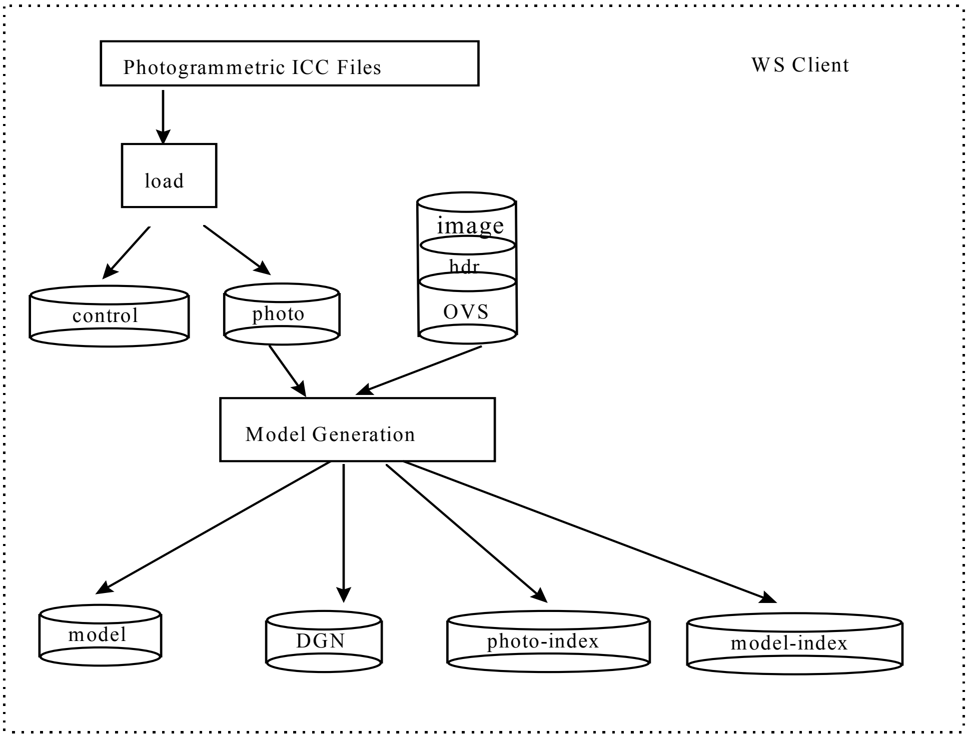
Two utilities are available for:

- printing a sketch of the measured points of a photo on a DIN-A4
- creating a photoindex by extracting the 1 MB overview from the image pyramid of each photo in the block. This photoindex can be displayed on the screen with Intergraph or MicroStation software and/or plotted on a HP DIN-A0 plotter. Each thumbnail is 15x15 cm on the plot.

5. Preparing Data for stereoplotting and DTM generation.

First, the *photo* file is updated with the adjusted photo coordinates of the pass and control points. In addition, we transfer from the GeoTex the external orientations and the self-calibration parameters (there is more information such as the covariance matrix of each observation and the reference systems used, but unfortunately the *photo* file cannot store it)

The software requires also a *control* file with the ground coordinates of the points and a *model* file containing an entry for each model. Each entry stores the identifiers of the images forming the stereopair and the collinearity parameters of the model.

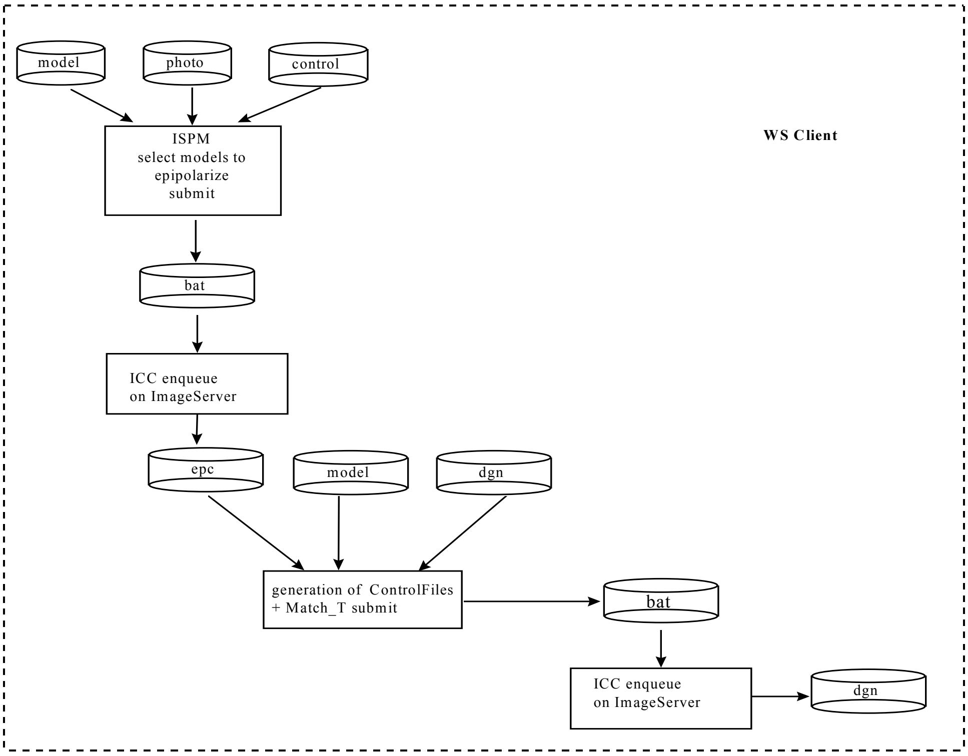


Again, the operator can use the standard GUI to define the stereopairs or our utility that creates entries for all the possible models with overlap between 50% and 85%, the percentages defined by the operator. Optionally, a DGN per model (the stereoplotting canvas) can be automatically created. Another one can be created with all the pass and control points for checking during stereoplotting.

6. Queuing Epipolar + Match_T ImageServer process.

The first step is to have the model sampled into epipolar geometry. We use the standard GUI to create a file containing the command lines for the epipolar process. With the BatchManager software this file can be enqueued to run in the server, with all the image files mounted locally.

For launching Match_T we do not use the standard GUI either. Instead, we prepare the required control and parameter files automatically and enqueue it for execution on the server.



The results of Match-T are stored in a DGN file. The 3D visual quality control is done with all the files mounted remote on the server.

The last improvement in our workflows has been reducing the time spent in exchanging execution reports among production units. We have tried to standardize these forms as much as possible and have improved the error reporting, to be aware of errors as soon as they appear.

7. Final Remarks.

As the time goes the amount of data grows larger and the hardware changes at a high rate. This leads to a continuous development effort and therefore systems and software have to be flexible to absorb the changes easily. This means easily extensible and changeable, fast traceable for debugging, and portable among environments. We would argue that the Object Oriented Programming is becoming the best development platform to meet these requirements.