A digital photogrammetric workstation on the WEB

P. Drap a, P. Grussenmeyer b, *

a GAMSAU UMR CNRS 694, Ecole d’Architecture, 184 Avenue de Luminy, Marseille F-13288, France

b ENSAIS-LERGEC, 24 Boulevard de la Victoire, Strasbourg Cedex F-67084, France

Received 1 December 1998; accepted 21 December 1999

Abstract

The main features of a WEB-based tool for digital architectural photogrammetry called Architectural Photogrammetry Network Tool for Education and Research (ARPENTEUR) are presented. The aim of this project is to propose a photogrammetric software package made freely available on the Internet as an applet through a simple browser (http://www.arpenteur.net). All the photogrammetric adjustment and image processing classes are written in Java™. The use of large applets in photogrammetry has not been very advanced until now but rate transfers and compatibility levels of Internet browsers will increase in the near future. Users can set up their own projects by transferring their images, camera, and control information on the ARPENTEUR servers from any place in the world connected on the Internet. Various examples of small format architectural photogrammetry projects are also accessible via WWW and can be used for teaching. This software is an extension of the Traitement d’Images et Photogrammétrie Numérique (TIPHON) software developed at ENSAIS-LERGEC. The photogrammetric model is currently computed by the traditional steps of inner, relative and absolute orientations. The measurements in the images can be performed manually or by image correlation. The measured points, lines, and geometrical primitives are recorded in text files and visualised inside a standard WEB browser. The plotting module is especially dedicated to architectural surveys and based on a formalisation of architectural and geometrical knowledge. The latest developments of the Java programming language and object-oriented systems are used in ARPENTEUR, both in the photogrammetric and in the architectural part, and ensure a high level of compatibility with any hardware platform supporting a WEB browser. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: digital photogrammetry; object-oriented technology; Java™; knowledge representation; WEB technology; education; architecture

1. Introduction

Most of the photogrammetric applications are used locally on individual computers and each application has to be installed separately. Some packages are operational within local networks or workgroups. The idea developed in the Architectural Photogrammetry Network Tool for Education and Research (ARPENTEUR) project is to propose a photogrammetric software package made available on the Internet as an applet (Fig. 1) through a simple browser. This project is managed by two French research groups located in Strasbourg (“Photogrammetry and Geomatics”) group at ENSAIS-LERGEC and in Marseille (Gamsau-MAP CNRS laboratory). In pho-
Fig. 1. ARPENTEUR’s project synoptic scheme, from Drap et al. (1999).

photogrammetry, the use of applets available on the Internet instead of local applications is, in most cases, not advanced. Applets are, most of the time, just used either to present examples on the Internet or to show methods of data processing. An interesting applet presenting image matching of réseau crosses has been presented by Höhle (1998). As another example, national mapping agencies often use applets for the computation or the transformation of geodetic coordinates.

Our research is motivated by different reasons.

• Software for education and research: developing software using Java™ is particularly convenient, when teams working in several places use different platforms (here Windows NT and Unix platforms located in Strasbourg and Marseille).

• A WEB tool dedicated to architectural survey: ARPENTEUR has the benefit of the two partners’ laboratory expertise in the field of close-range photogrammetry and architectural knowledge representation in a survey process. This collaboration is enriching for both researchers and students working on the project.

• A simple photogrammetric system for archaeologists, architects and photogrammetrists available wherever they are in the world. As an example, the members of International Committee for Architectural Photogrammetry (CIPA) can benefit from the use of a digital photogrammetric workstation on the WEB for simple projects and presentation of examples on the ARPENTEUR servers.

The purpose of the paper is to present the architecture and the capabilities of ARPENTEUR based on the latest developments of Java programming language and object-oriented systems.

2. The use of object-oriented technologies

2.1. Object-oriented technologies and knowledge representation

The strategy developed for ARPENTEUR, both in the photogrammetric development and in the architectural approach, is directed by data-processing object-oriented languages regarding the formalisation of complex knowledge fields.

The object-oriented formalism is, to some extent, the translation into computer language of the reason-
ing by classification. The universe of knowledge can be divided into elementary objects structured by specialisation of properties. With this approach, we can model complex phenomena by the description of the relations between the various objects and reach a high level of abstraction. The object-oriented programming languages give an opportunity to formalise and handle models of complex knowledge fields. The field of knowledge is split into elementary concepts, structured through refinements of classes (Oussalah et al., 1997).

2.2. The Java environment

The programming language chosen for the ARPENTEUR software is Java. There are at least two reasons for this choice:

- Java is strongly object-oriented; and
- Java is a platform-independent programming language supposed to run anywhere and supported by a lot of Internet browsers.

Java is a recent language, based on OAK developed by Grosling at SUN in 1991. OAK was an object-oriented language and platform-independent. In 1995, OAK became Java by addition of Internet capabilities. Java is given by SUN to the Internet community with its compiler, library, specifications, and documentation. The code written in Java is compiled in order to produce byte-code, which is a low-level format. This resulting code is transferred locally during the loading of the applet and interpreted on each different hardware by the Java Virtual Machine. For each computer, the Java Virtual Machine will be different, that is why Java is independent of platform.

2.3. The Model–View–Controller (MVC) paradigm

Building a modular object-oriented user interface is a very hard task. The result is difficult to maintain, complex to understand, and often not reusable. However, there is a lot of benefit to separate the user interface from the application code. This kind of separation in an object-oriented context is often made using the MVC method. The MVC, originated in Smalltalk, is used in many different contexts. Java, with its event delegation model, allows us to use it in this application (Fig. 2). The advantages of this approach allow:

- a methodical approach of Graphical User Interface (GUI) design;
- an independent development of application components (more than one programmer at the same time is possible);
- the reusability of the application and the user interface components; and
- an easy development of new features.

In this context, independence between interface and model means that different interfaces can be used by the same application. This means that the application does not know which interface is currently connected to it. Similarly, any part of the system can be changed without causing any disturbance to the others.

2.4. The photogrammetric objects

The photogrammetric universe is modeled as an organisation of concepts in different hierarchies and relationships to each other (Fig. 3). In addition, the

---

**Fig. 2.** MVC synoptic scheme in ARPENTEUR.
Fig. 3. The hierarchy of the MVC objects.

3. Object-oriented database management system

3.1. Description of network database process architecture

ARPENTEUR uses an Object-Oriented Database Management System (OODBMS) which provides an Application Programming Interface (API) for storing and sharing Java objects among users, hosts, and programs. The user can modify a subset of his own data through the network (e.g., update a measurement in the relative orientation process). The object storage is made on the server and monitored by a Remote Method Invocation (RMI) Java Server directly connected with the local client. This RMI server, which allows the objects to move across the network, is the interface between local objects available on the client computer and the database located on the main server.

3.2. Managing objects with an OODBMS

The object storage management used in ARPENTEUR is performed by the OODBMS and has one distinctive feature: the programming language is the same as the language used to manage the data. That means that there is no difference between describing a system (the photogrammetric process in our case) using an object-oriented approach in Java and elaborating the database scheme.

This is particularly convenient and makes the database easy to use. A lot of concepts used to
describe the photogrammetric process are dependent on each other; e.g., in Fig. 4, the ‘photogrammetric model’ object uses a set of ‘images’ which uses a camera model and so on. The object-oriented approach guarantees that saving a model also means saving all its dependencies and, in addition, saving a part of a model, i.e., a new measurement in an orientation phase is made by a simple update without re-storing all model data.

4. A digital photogrammetric tool on the WEB

The photogrammetric approach is based on well-known algorithms (Kraus, 1997a,b) used in the Traitement d’Images et Photogrammétrie Numérique (TIPHON) software (Grussenmeyer and Koehl, 1998). The standard steps include inner, relative and absolute orientation. ARPENTEUR is developed in Java, using JDK 1.1.7, and is operational from any hardware platform supporting a WEB browser using this level of Java. Presently, the software is tested with Netscape Communicator 4.51 on PC and Netscape Communicator 4.6 on SUN workstations.

4.1. Internet access

The software is accessible via the Internet network and a WEB browser (Fig. 5). We use three kinds of servers to offer various services. Using ARPENTEUR means three steps.

4.1.1. HTTP server

The HTTP server and the servlet module (a servlet module runs inside request/response-oriented servers, and provides a way to generate dynamic html documents) allow the creation of a workspace by a registration page. The user has access to an example project about an aqueduct in Cairo (Grussenmeyer and Abdallah, 1997). Once a user has registered, he has free access to the whole capability of ARPENTEUR.

4.1.2. FTP server for the use of ARPENTEUR with an own project

Of course, a user can use his own project. He has a specific access to his workspace via an FTP server directly accessible through the main windows. The FTP server allows to read files saved on the user
workspace, so he can put images, camera definition files, and ground control point files on the server and get the output text files generated by the system.

4.1.3. RMI server

ARPENTEUR is an applet using a lot of Java classes. That means that the process is running on the client computer; the Java security restrictions do not allow an applet to read or write files on the client disk. To solve this problem, we have developed a Java application server, running on the server machine and dedicated to saving files on the server disk and managing objects through the network using the RMI method. Each connected client has his own RMI server to ensure data integrity. Thus, all files (camera, control points, etc.) and images of each project are stored on the server, but are password-protected, such that only the user and the system administrator have access to them. To solve local access problems, we are working on another version to be used on an Intranet network.

Two servers located in Marseilles and Strasbourg are currently dedicated to this project. Due to storage limitations, the image size is currently limited to 2 MB. All the result files, the orientation reports, and the plotting files are written in the user workspace on the server by this Java application server after connection with the client.

4.2. The photogrammetric functions

4.2.1. Orientations

The photogrammetric development follows the TIPHON software developed at ENSAIS (Grus-
senmeyer et al., 1998). In addition to the standard orientation (Fig. 6) steps on analytical or digital stereopotters, we also propose an absolute orientation made without control points. We define a local coordinate system by measuring particular points on the model (vertical lines, wall corners), and a scaling of the model by measuring some distances on the object. The goal is to provide an easy-to-use photogrammetric software package, allowing also simple photogrammetric surveys with non-metric cameras and without theodolite.

4.2.2. Measuring points

For point measurement, we use image correlation in the inner orientation (e.g., automatic measurement of réseau crosses) and also for the measurement of homologous points (during the outer orientation and the plotting process). The used method is a combina-

![Fig. 6. ARPENTEUR’s relative orientation frame, from project presented in Grussenmeyer and Abdallah (1997).](image-url)
4.2.3. Geometrical tools

In order to propose a photogrammetric tool dedicated to architecture, we added some geometrical tools to manage point measurement on geometrical primitives (Florenzano et al., 1997). A set of least square computation models is available like computation of lines, planes, circles, and cylinders using 3D measured points on their surface.

4.3. Graphical and text output results

The results of all orientation steps are recorded in an ASCII file. Therefore, it is often difficult to have a good visualisation of numerical results just by looking at numbers. To represent the 3D data generated by the model computation, we use the 3D file format of MicroStation (Bentley™). We also developed a set of routines able to write geometrical information in this file format. Thus, at each orientation step, we create a file in this format directly readable by the MicroStation software. For example, after the inner orientation of a réseau camera, the system generates automatically a DGN file, which represents the discrepancies between theoretical values of the réseau and measured crosses. A scale factor is computed to represent the discrepancies as visible vectors on the screen. If the client does not use MicroStation, he can get the results in the ASCII files or use the Virtual Reality Modeling Language (VRML) output (see Section 4.4).

4.4. VRML output

VRML is a platform-independent interpreted language dedicated to the visualisation of 3D scenes inside a standard WEB browser. All objects manipulated by ARPENTEUR which implement a VRML interface are enabled to visualise their instance created through the survey process. This allows a direct monitoring of the scene from within the applet. Measured points, controls points, geometrical primitives, and architectural objects can therefore be visualised during the successive steps of the survey (once the absolute orientation is done). Java/VRML interaction is used in order to allow the user to retrieve data from an object in the scene by a mouse click on its VRML representation. We are also working on a Java visualisation tool avoiding the installation of an additional VRML plug-in in the browser.

5. Present problems with Java and internet

The use of Java and the WEB platform is full of advantages and promises. Distributing software on any platform at any place connected to Internet is, of course, very interesting. The updating is done directly on the server and the user always works with the latest release of ARPENTEUR. But presently, we have to deal with three delicate points by proposing ARPENTEUR on Internet.

5.1. Loading time

This software architecture is based on client–server relation: the software is loaded through the network and runs on the client computer. The loading time can be important; the software itself is a large file and the loading time also depends on the data size since all data, and especially image files, are stored on the server. In addition, the data transfer rate on the network is not regular. Therefore, once the files and images are loaded on the remote computer, the use of the disk cache makes it possible to work properly.

5.2. Instability of programming language

Java is a recent language in constant updating. We always choose to work with the latest versions (currently JDK 1.1.7) but this leads to some problems.

- The compiler is not stable and a lot of bugs still exist.
- We do not know on which WEB browser the applet will run, and usually most of the browsers are one or two versions behind the latest version of the Java Virtual Machine. Using the latest version of Java means that a very limited number of people connected can use our tool. This is in apparent contradiction with the goal: large public access. We rely on future developments of Java.
and its diffusion around Internet to improve this point.
• Presently, all the Java Virtual Machines are not at the same level, so it is possible that some functions do not run on certain platforms.
• Since the first version of ARPENTEUR presented at the ISPRS Commun. 5 Symposium in Hakodate (Drap and Grussenmeyer, 1998), we have chosen the strongly MVC-oriented Swing API provided by Sun.
• We plan to use the latest version of JDK (1.2) at the beginning of 2000. Working with a newer version of Java is more comfortable for developing software; therefore, it poses restrictions on the software use. Actually, the Internet browsers do not run with the Java Virtual Machine 1.2 and the future users will have to first download the necessary Java plug-ins from SUN before using the ARPENTEUR applet.

5.3. Execution time

Java is compiled in byte-code, then interpreted by the Virtual Machine. This allows multiple computers running the same software but increases the execution time, since the applet is running within the browser. Presently, executing software in Java takes 10 or 20 times longer than in C++ native code. Just-In-Time (JIT) compilers, which are faster, are not possible with ARPENTEUR that has to be used with the Java Virtual Machine of the browser. This is a problem for computer-intensive operations like image processing. This situation may be improved in the future.

6. Perspectives: a photogrammetric tool dedicated to architecture

The ARPENTEUR project is relatively new. The development started at the end of 1997 and now a version of a standard photogrammetric process is available (Grussenmeyer et al., 1999). We organise our future work in three directions:

• to increase the reliability and interface user-friendliness of the photogrammetric tools;
• to add photogrammetric functions in the orientation process (i.e., bundle adjustment); and
• to develop a tool dedicated to architectural survey.

6.1. The photogrammetric software

We are working on improving the photogrammetric software performance, the didactical reports, the speed, and the robustness of algorithms. The use of Internet as an access platform will enable us to offer multi-language access to ARPENTEUR. In 2000, English, Italian and Spanish versions will be available. The ARPENTEUR project, presently managed by ENSAIS and GAMSAU, can accept in the future other contributions from the international community. As an example, the kernel of the Java visualisation tool recently implemented is a contribution of T. Landspurg (Landspurg, 1999).

6.2. A tool dedicated to architectural survey

Our first objective is to dedicate the plotting module to architectural applications. This module is based on a formalisation of architectural and geometrical knowledge. The architectural measured object must be first identified and structured. Then, the architectural knowledge is used to guide and control the measuring process of a building. This means that the morphology of the measured object is already known (Florenzano et al., 1997; Drap et al., 1999). Elements of a building are described as “entities” (elementary objects) providing that they meet two requirements:

• an entity is a unique “object” identified by a single element of the architectural vocabulary; and
• an entity has an obvious and permanent role in the physical structure of the building.

The benefit of a tight link between knowledge and measurements will appear at several levels:

• The survey process of architectural entities can be drastically simplified and limited to a quick control of the underlying model, if desired; and
The photogrammetric tool allows the user to both validate a structural architectural model and study any surface alteration.

The use of an object-oriented approach allows us to increase the set of architectural entities available in the system. Actually, this phase requires the knowledge of Java programming, but this operation should be able to be performed by a user without high computer science knowledge. To solve this problem, we are currently developing an additional tool named ‘Classeur’ able to generate Java source regarding a textual definition of a new entity. This tool will compile the source on the server and load the new class on the client computer. This development will give a dynamic aspect to the system.

7. Conclusions

ARPENTEUR is presently accessible via Internet (http://www.arpenteur.net) and allows basic photogrammetric functions like inner and outer orientations and different survey functions. This project, which only started at the end of 1997, will be improved by the extension of a photogrammetric software library to offer our students and researchers a development platform and tools for their projects combining photogrammetry, computer science, architecture, and archaeology. We try to use the advantages of the interdisciplinary context of our project to develop a photogrammetric tool dedicated to architectural survey. Presently, ARPENTEUR is used for teaching photogrammetry with small-format images to students at ENSAIS and GAMSAU. Even if the use of large applets on Internet is not always easy, we are confident in the future of Java and Internet and hope that rate transfers and efficiency of personal computers, as well as the compatibility levels of Internet browsers, will increase soon to make such a project a realistic one. Finally, we think that this project will meet the CIPA objectives by offering this community a free photogrammetric tool for simple projects. The integration of architectural knowledge into the software is in accordance with the paradigm of architectural survey: increasing knowledge in surveying so that surveying might broaden our knowledge.

Acknowledgements

The authors wish to thank P. Capron (Strasbourg) and G. Gaillard (Marseille) for their contribution to the development of the ARPENTEUR software package in 1999 during their postgraduate studies at ENSAIS and GAMSAU.

References

