UPTK: A FREE TOOLKIT TO SUPPORT COOPERATIVE RESEARCH IN PHOTOGRAMMETRY

Antonio Maria Garcia Tommaselli, Prof. Dr.
Júlio K. Hasegawa, Prof. Dr.
Maurício Galo, Prof. Dr.
São Paulo State University, Department of Cartography, Pres. Prudente
Rua Roberto Simonsen, 305
19060-900 Presidente Prudente, S.P., Brasil
{tomaseli, hasegawa, galo}@prudente.unesp.br

Summary

This paper presents the development status of the Unesp Photogrammetric Tool Kit (UPTK) as well as its main features. This library is a tool kit designed to support photogrammetric research with a set of functions and classes written in C and C++ languages. The library is organized in six modules: UPTK_ALG, operations with vectors, matrices and parameters estimation; UPTK_FEXT; functions for feature extraction; UPTK_FOTOG; photogrammetric models and processes; UPTK_GEOM; coordinates transformation; UPTK_IMG; low level processing and image manipulation; UPTK_PRIM; general purpose functions. Some modules have versions both in C and C++ languages but most of the source code has different implementations in C and C++. The source code was documented using Doxygen, which enabled the automatic generation of html pages. Three different binary files for three different compilers were released for download. Besides the functions and classes, some photogrammetric tools are also available at the web site of the project, like a camera calibration package. A set of images collected with small format digital cameras can also be downloaded at: www.prudente.unesp.br/dcartog.uptk. A general view of each module and the implemented algorithms will be presented with some details of the most relevant files.

Keywords: Photogrammetry, software development, C/C++ language.

1. Introduction

The development of a toolkit to support teaching and research in Photogrammetry was a need detected by the Group of Photogrammetry, a research staff dedicated to teaching and research in the São Paulo State University, campus at Presidente Prudente, São Paulo. Some main reasons that motivated the development of the library were:

- The researchers in Photogrammetry and Geodesy usually write most of their source code in several distinct languages (Fortran, Pascal, C and C++) but there is a little effort for code reuse;
- Due to the nature of the research and rigid project schedules, researchers tend to leave to a second level of importance the use of good programming practices. There is a belief that is more productive to spend time in developing and testing new algorithms instead of acquiring new programming skills or implementing more detailed tools;
- A low level of code reuse between local researchers is verified, which represents a high cost for scientific research. In several research projects, time is spent implementing already existing tools, when that could be instanced from the existing one. As a consequence, there is a lot of code overlapping due to the lack of public code sharing and suitable documentation. Code reuse is not a systematic practice and it is dependent of several changes in the original program, which often requires the assistance of the original programmer;
- Concerning the Unesp Photogrammetric Research Group, after more than 10 years of activity, any software was registered or commercially licensed, although several thesis, dissertations and papers were produced using those developments;
- Any standard was used for code documentation, being left to the researcher the decision of what style to use and how to document the program. Besides that, several old programs were written without attention to the efficiency and computation cost.

Several initiatives were taken in order to produce a library of programs, but none were continued [4]. In 1999, as a result of several projects ([2], [3], [7]), the UPTK project was firstly conceived but it was just formally proposed in 2001. A great impulse to the development of the UPTK library was the SEIRA project, a joint international project supported by the IBEROEKA program, with Spanish (Institut of Geomatics and Stereocart) and Brazilian (Unesp and Aerocarta) partners. The aim of the SEIRA [9] project was the implementation of an experimental prototype to test the concepts of “lightweight APRS and fast mapping” using medium format digital cameras and a low cost, yet precise, navigation system. It became clear for the Brazilian research team at Unesp, that the existing computing programs should be organized in a public library and that the next developments should include a standard documentation procedure, otherwise it would be unfeasible to participate in more complex research projects. Those were the main motivation to start the UPTK project, which received the financial support of Fapesp (Fundação de Amparo à Pesquisa do Estado de São Paulo).

The aim of the UPTK project (Unesp Photogrammetric Tool Kit) is the implementation of a set of functions and classes written in C and C++ languages to support research in Photogrammetry. Besides this main goal, other aims can be mentioned: to test each module and include example programs; to provide data sets to test the modules and to enhance teaching activities; to share the developed libraries with other research and teaching institutions; to learn and use other public domain tool kits.

2. Programming language

The proposal of augmenting code reuse within the research group will be effective if a minimum set of rules and procedures is adopted. The first difficult is the variety of programming languages and styles often used. Fortran, C, C++ and Pascal languages have been used and it is a hard work to convert these different source codes to a common language. Even when using the same language, there are variations due to the programming environment, e.g., when using Borland graphics facilities, some programmers tend to use the proprietary data structures throughout the code, making the conversion to other compilers a boring task. Besides that, as the researchers usually write their own codes, recommended programming practices [5] are often left aside because it is often considered more important to test an algorithm than making a well structured and documented code. After several discussions it was decided to use both C and C++ languages and convert Fortran routines and Pascal procedures or implement it again. There was a line of thought that put some restrictions to the use of C++ in basic functions, due to some tests in which it was verified some loss of efficiency when using several levels of abstraction. Siek and Lumsdaine [6], however, tested their Matrix Template Library, and showed that their approach for algebra generic programming does not hinder high performance.

Despite the fact that most of the researches are more familiar with C programming, there is no doubt that C++ is more suitable when considering code reuse and great projects. The solution for this diversity was the adoption of common programming and documentation rules, making feasible code reuse.

3. Other tks and libraries

The primary aim of the UPTK project is to produce a set of functions to support research in Photogrammetry. Originally, it was conceived as an internal initiative, but some researchers of other institutions asked the permission to use the libraries and it was decided to freely distribute the binaries. Before that, a seek for generic libraries was done and several tools were find out and will be mentioned briefly.

One of the best examples of public libraries in the Geomatics field of knowledge is **TerraLib** [20], a GIS classes and functions library, available from the Internet as open source, allowing a collaborative environment and its use for the development of multiple GIS tools. Its main aim is to enable the development of a new generation of GIS applications, based on the technological advances on spatial databases [20].

An example of free source code for Photogrammetry, aiming the development of an educational digital photogrammetric workstation is the **Efoto** project. This project aimed at to establish a line of research involving the computational development of digital photogrammetry routines, using a high-level programming language [1].

Graphic user interfaces (GUI) can be implemented using proprietary solutions or free code. Some examples of free graphics packages are the Qt and the GTK. **QT** is a multiplatform C++ GUI application framework. It provides application developers with all the functionality needed to build applications with state-of-the-art graphical user interfaces. Qt is fully object-oriented, easily extensible, and allows true component programming. There are free versions of Qt that can be used under the GNU public license [19]. **GTK** (GIMP Toolkit) is a library for creating graphical user interfaces. It is licensed using the LGPL license; open software, free software, or even commercial non-free software can be developed using GTK [14]. Programmers using OpenGL can access **GLT** (Open GL), a C++ class library for programming interactive 3D graphics with OpenGL [13].

Several examples of packages for image processing and computer vision are available in the web, and some of them are mentioned here. **Gandalf** is a computer vision and numerical algorithm library, written in C, which allows the development of new applications, distributed under the Lesser Gnu Public License [12]. **VXL (Vision-Something-Libraries)** is a collection of C++ libraries designed for computer vision research. It was created from TargetJr and the Image Understanding Environment (IUE) with the aim of making a lighter, faster and more consistent system. VXL is written in ANSI/ISO C++ and is designed to be portable over many platforms [23]. **LIMP (Large Image Maniputation Program)** was started as a platform for testing new techniques for image processing. A lot of complex code can be required to make an efficient library for applying arbitrary computations to large datasets ("large" in this case meaning too big to fit into memory). The goal of LIMP was to move as much complex code as possible into the library, leaving a simple yet powerful way of dealing with images from a user's code [16]. The **TINA (Open Source Image Analysis Environment)** algorithm development libraries have been developed over a number of years to simplify the task of vision algorithm development and evaluation [21]. The **PVT (Projective Vision Toolkit)** is a series of utilities that allows one to take an image sequence and compute the fundamental matrix and trilinear tensor. This can be used for such problems as camera self-calibration, structure from motion, camera motion annotation, image stabilization, 3D tracking and recognition, etc [18]. The **NetPbm** is a popular set of graphic programs and programming libraries. There are over 220 separate programs in the package, working with the netpbm formats, in several different categories, e.g, image converters, image generators, image editors and image analyzers [17].

Other packages can be mentioned like the **GaMa** (**Geodesy and Mapping C++ Library - GNU GaMa/GaMaLib**), a C++ free software for adjustment of geodetic free networks released under GNU General Public License and the LINPACK, a collection of Fortran subroutines that allows the analysis and the solution of linear systems and linear least-squares problems.

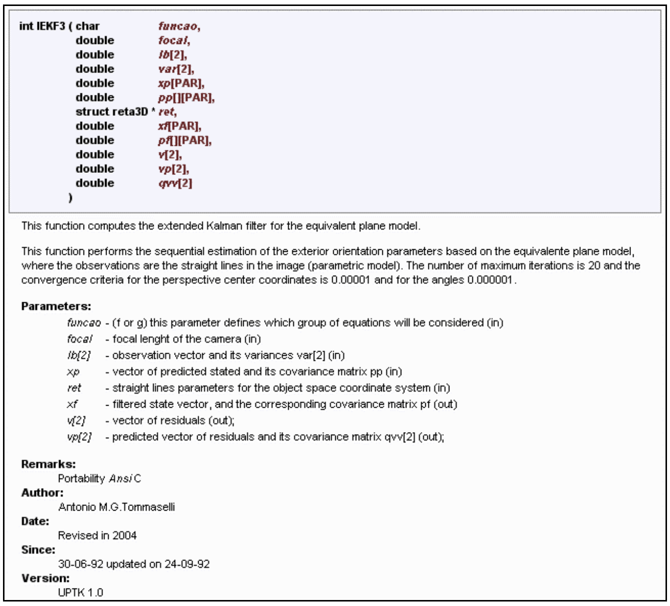
4. Documentation

One of the critical issues in developing shared programs is the documentation, mainly when the team has programmers with different levels of expertise, from undergraduate students to senior scientists with experience in traditional languages.

The original approach of the UPTK development team for documention was to produce just commented source code, but after of some months of work it became clear that this strategy would be unsuccessful. A search for automatic tools for generating manuals led to several packages, some of them freely distributed: AutoDOC, Cocoon, CcDoc, Cxref, Doxygen, HTMLgen, ManStyle, ScanDoc.

```c
#include <uptk\c\fotog\uptk_iekf_slines.h>
#include <uptk\c\alg\uptk_matrizes.h>

/**
  \brief This function computes the extended Kalman filter for the equivalent
         plane model

         This function performs the sequential estimation of the exterior
         orientation parameters based on the equivalente plane model, where the
         observations are the straight lines in the image (parametric model).
         The number of maximum iterations is 20 and the convergence
         criteria for the perspective center coordinates is 0.00001 and
         for the angles 0.000001.

  \param  funcao - (f or g) this parameter defines which group of equations
          will be considered (in)
  \param  focal - focal lenght of the camera (in)
  \param  lb[2] - observation vector and its variances var[2] (in)
  \param  xp - vector of predicted stated and its covariance matrix pp (in)
  \param  ret - straight lines parameters for the object space coordinate
          system (in)
  \param  xf - filtered state vector, and the corresponding covariance
          matrix pf (out)
  \param  v[2] - vector of residuals (out);
  \param  vp[2] - predicted vector of residuals and its covariance matrix
          qvv[2] (out);

  \remarks Portability \a Ansi C
  \author  Antonio M.G.Tommaselli

  \date    Revised in 2004
  \since   30-06-92 updated on 24-09-92
  \version UPTK 1.0

*************************************************************************** */

  int IEKF3 (char funcao, double focal, double lb[2], double var[2],
             double xp[PAR], double pp[][PAR], struct reta3D *ret,
             double xf[PAR], double pf[][PAR], double v[2], double vp[2],
             double qvv[2])
 {
```

(a)



(b)

Figure 1. Documentation using Doxygen: (a) a fragment of a source code in C; (b) resulting documentation in HTML viewed with a web browser

The Doxygen package was chosen due the easy of use and other interesting features [10]. Doxygen is a documentation system for C++, C, Java, IDL (Corba, Microsoft, and KDE-DCOP flavors) and to some extent PHP. It can generate an on-line documentation files in HTML and/or an off-line reference manual from a set of documented source files. There is also support for generating output in RTF, XML, Latex and other formats. The documentation is extracted directly from the sources files, which makes it much easier to keep the documentation consistent with the source code. Doxygen can also be configured to extract the code structure from undocumented source files.

Figure 1 presents an example of how to use doxygen commands. In Figure 1.(a) a fragment of a source file in C is presented, showing the special commands inserted. Figure 2.(b) presents the resulting manual, visualized in a web browser.

5. UPTK Structure

The organization of the UPTK library was done in several steps. Firstly, a rough estimate of the amount of programs and its functionalities was generated. Then, the source code files were collected and it was verified its usability and the necessary changes. The files were documented using the Doxygen commands and some examples were produced. The files that required minor changes to use as common functions were adapted. In certain cases, however, it was unfeasible to modify the program to use common libraries, because of the algorithms used. An example of this are the matrix manipulations functions used in bundle adjustment functions. As a consequence, some redundancy still exists, as the users will notice. For example, there are two files for matrices and vector algebra and two functions implementing rotation matrices and collinearity equations. Although the models are the same, the impact in the source code does not worth the adoption of a unique function. For next versions, this redundancy will be minimized.

The UPTK library is organized in six modules, with implementations in C and C++ languages. Just a few modules are implemented both in C and C++, e.g., matrices algebra. Most of the files are different in C and C++. Table 1 presents the general description of each module.

Module	Description	Languages
UPTK_PRIM	General purpose primitive functions	C and C++
UPTK_ALG	Matrices and vectors algebra and parameter estimation	C and C++
UPTK_IMG	Image access and low level processing for enhancement	C and C++
UPTK_FOTOG	Photogrammetric models and processes	C and C++
UPTK_FEXT	Feature extraction	C and C++
UPTK_GEOM	Geometric modeling and coordinate transformation	C

Table 1. Modules of the UPTK library.

A detailed description of all modules and its data structures, classes and functions is unfeasible to present in this paper, due to the lack of space. Details can be obtained in the UPTK project web page. A brief description of the modules and its files, with some highlights will be presented following.

Uptk_C_Alg Module

This module have two files: uptk_alg.c and uptk_matrizes.c. Both have functions for matrices and vector operations, but with different implementations. In the uptk_alg file the matrices are stored as vectors, enabling some optimizations, but this approach requires some address arithmetic. Examples of functions in this module: product, sum, and other operations with matrices and vectors, matrices inversion, Choleski decomposition, rotation matrix using quaternions.

Uptk_C_Fext Module

This module holds several functions that can be used for image analysis and feature extraction. There are two files (uptk_feat_ext.c and uptk_fext.c) with functions that performs operations like: computation of gray level gradients; thresholding, thinning with non-maxima suppression; edge labeling and linking; line fitting; corner detection with Moravec operator; center of circular symmetry.

Uptk_C_Fotog Module

This module stores several conventional and experimental photogrammetric model, e.g., functions for conventional space resection, bundle block triangulation, space resection with straight lines. Details of the files are presented in Table 2.

uptk_fotog.c	Basic models used in Photogrammetry like: rotation matrix, collinearity, essential matrix, fundamental matrix, coplanarity, 3D intersection.
uptk_iekf_slines.c	Functions implementing a sequential approach for space Resection using straight lines and the IEKF.
uptk_intersec.c	Functions that compute 3D coordinates using photogrammetric intersection.
uptk_kfi_slines.c	Functions implementing a simultaneous approach for space resection using straight lines and conventional least squares.
uptk_resp.c	Conventional space resection with control points.
uptk_simil_3D.c	3D Helmert transformation with parameter estimation using least squares.
uptk_transfor.c	Functions for computing parameters of 2D geometric transformations using least squares.
ft_aux.c	Functions for triangulation with bundle block adjustment.

Table 2. Description of the files in the uptk_fotog module.

Uptk_C_Geom Module

The functions of this module were originally written in Fortran and the conversion to C/C++ is still under development and at the moment there are functions to transform geographic to TM – Transverse Mercator coordinates.

Uptk_C_Img Module

This module has a file with functions for image storage in pbm and related formats. Some basic low level operations with images will be added to this module.

Uptk_C_Prim Module

In this module several data type definitions, macros and basic mathematical operations are defined. Basic messages are also defined in this module.

Uptk_Cpp_Alg Module

This module has a C++ implementation of a class **UPTKGMatrix** designed to handle generic data dynamically. This class of objects performs matrices operations like: sum, subtraction, product, inversion, transpose. Besides this generic matrix implementation, a second file has conventional matrices operations, similar to those functions in the Uptk_C_Alg module.

Uptk_Cpp_Fext Module

In this module there are two groups of functions: functions for image segmentation, like region growing and boundary following, which were encapsulated in a class. These operations are applied on gray level images, stored in a class named UPTKImageGray; the second file holds functions for analysis of reference windows aiming at the process of gray level correlation. These functions compute the signal variance, the noise variance and the covariance matrix of the translation between the reference and the search window, which avoids homogeneous and edge regions for correlation.

Uptk_Cpp_Fotog Module

This module stores several files that performs operations like: epipolar line search; interior orientation for digital images including methods for correcting lens distortion and photogrammetric refraction; epipolar resampling; orientation with a modified coplanarity model. Several data structures were designed to store the exterior and interior orientation parameters.

Uptk_Cpp_Img Module

This is a complex module with several implementations for image manipulation. In Table 3 the functionalities of some files of this module are described.

uptk_correlacao.cpp	Functions for gray level correspondence: cross correlation, correlation coefficient, error function and modified error function.
uptk_filters.cpp	Functions for image filtering: some operations are applied only to gray level images, stored in the class UPTKImageGray, whilst others operations in both gray and RGB images, which are stored in the class UPTKImage. The functions available are presented in Table 4.
uptk_image.cpp	Implementations of the class UPTKImage; handles gray level and RGB images. The class UPTKImage has either an element UPTKPixelGray or UPTKPixelRGB. Each element of the object UPTKImage is stored in public objects; they can be Gray (a gray level image of the class UPTKImageGray) and RGB (a RGB image of the class UPTKImageRGB). Several functions control operations like loading and saving images, inversion, conversion from RGB to HLS, image rotation and pyramid generation.
uptk_image_gray.cpp	Implementations of the class UPTKImageGray.
uptk_image_rgb.cpp	Implementations of the class UPTKImageRGB.
uptk_imageview.cpp	Functions for image visualization using the Timage object of the Borland Builder environment.
uptk_mq.cpp	Implementation of the Least Squares imagem matching.
uptk_realce.cpp	Histogram stretch.
uptk_reamostragem.cpp	Implementation for image resampling. This class has two virtual methods: FMap e FMapInv, which are the direct and inverse mapping functions. To implement an image resampling, a new class have to be defined inheriting the attributes of this class, but implementing also the mapping functions, using the desired model, e.g., geometric transformations, polynomial models, collinearity equations, etc.

Table 3. Description of some files of the uptk_cpp_img module

Functions	Description
MediaFilter	Media filter for gray level images.
MedianFilter	Median filter for gray level images.
ThresholdLocal	Local thresholding for gray level images.
MedianAVarFilter	Edge preserving smoothing with local variance analysis..
OTSUThresholdGray	OTSU thresholding method.
PUNThresholdGray	PUN thresholding method.
ThresholdFilter	Global thresholding methods.
GaussianFilter	Gaussian filter for gray level or RGB images.
GenerateHistogram	Histogram of a gray level image.
ImageHistogram	Histogram on an image.
CEGordonRangayyan	Local contrast enhancement for gray level images.
EqualizeHitogramImage	Global histogram equalization.

Table 4. Functions available for filtering in the uptk_cpp_img module.

Uptk_Cpp_Prim Module

This module embraces several implementations for string manipulation, operations with generic lists, and definitions and implementations for memory handling.

Examples

Most of the implemented functions were documented using the Doxygen style and some example files were created in order to help the users to use the libraries. These examples are stored in a separate directory, but they can be accessed using the existing links within the documentation.

6. Distribution

As previously mentioned, the aim of the UPTK project (Unesp Photogrammetric Tool Kit) is the implementation of a set of functions and classes written in C and C++ languages to support research in Photogrammetry. These libraries were compiled with distinct compilers (Dev_C++, Borland Builder and MS Visual C++) and are available as static libraries in the project web page. The source codes are not available for download but can be requested by email either to the project manager or directly to the programmers.

There are three installations files corresponding to 3 different environments. Additional compilations can be provided upon request, as well as fragments of source code. The installation creates the directory structure presented in Figure 2. The root is the *include* directory of the compiler being used. Documentation and examples are also installed. The binaries can be moved to another directory according to the user programming practices. Although extensive tests have performed with the implemented tools, some bugs can still be found when using the libraries.

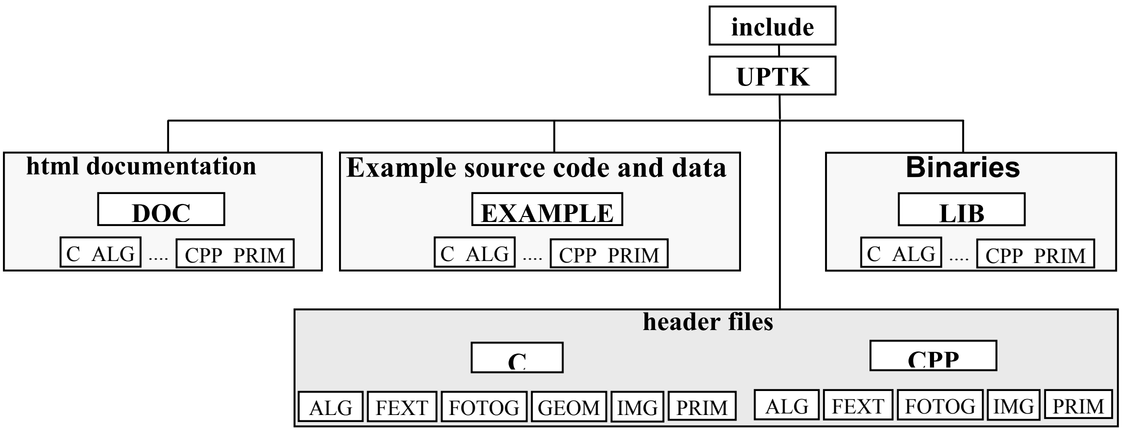


Figure 2. Directory structure created by the uptk installation file.

Besides static libraries and documentation, some photogrammetric tools and two data sets, with digital images acquired with non-metric digital cameras, are available for download.

Photogrammetric tools

Several photogrametric tools are available for download, e.g., software for camera calibration and stereoplotting. The CC (Camera Calibration) program is aimed at the calculation of the interior orientation parameters, such as focal length, principal point position and lens distortion. In order to achieve that, the coordinates of a set of ground points, have to be measured in distinct images. This free version is limited up to four images. It is recommended that the user read carefully the structure of the files with the options for processing. Some results can also be viewed graphically with the public domain Gnuplot software, which is included in this patch. A light softcopy plotter is available aiming at the photogrammetric compilation from images taken with digital cameras. Monoscopic or stereoscopic restitution (using a mirror stereoscope) can be performed with this software (Figure 3). Most of the available documentation of these tolls are still in Portuguese, but versions in English are being planned for the next releases.

Figure 3. Screenshot of a lite softcopy stereoplotter.

Image data sets

Non-metric digital cameras can be considered as an alternative for the compilation of georeferenced information, when the area to be mapped is not large and if the accuracy requirements are flexible. Two blocks of aerial images taken with non-metric digital cameras are available for download, as well as their exterior orientation parameters. These images were taken with two distinct cameras over the Unesp-FCT campus: Kodak DCS-420 and Kodak Professional 14N. The images can be freely used given the proper credits to the companies that collected the images and to UPTK project. The image blocks were triangulated using the Hats software of the Socet SET® 4.4.1, from BAE. The imaged area comprises the Unesp-FCT campus and its neighborhoods. The IR images were acquired in 2000 (DCS-420) and the RGB images in 2003 (DCS 14N). Both flights were kindly provided by the Photon Company. The earlier coverage was flight with the aircraft of the Multiespectral Company, and the acknowledgments must be given to those companies. Details can be found in the project web page.

7. Conclusion

Designing, implementing, and documenting a library for a specific application domain is a worthy challenge. Getting a library into widespread use within its intended user community is another major challenge involving teaching at many levels. UPTK is still in the first step because some drawbacks verified in the beginning of the project still persist. One is the duplication of data structures and implementations with the same functionalities; other is the adoption of similar programming practices by the programmers, which requires a change of research paradigms.

The UPTK project is being continuously developed, meaning that new versions will be released. It is expected that users report bugs and restrictions in the functions and classes, contributing for the improvements in future versions. For the forthcoming versions it expected to diminish the code overlap and to introduce new functions, e.g., for orthorectification and DTM generation. A great problem that should be approached is the documentation language. An effort should be done to provide documentation both in English and Portuguese.

More details about the project can be found on the web page:
http://www2.prudente.unesp.br/dcartog/uptk/.

Acknowledgements

The authors would like to thank the financial support of Fapesp (Fundação de Amparo a Pesquisa do Estado de São Paulo, grant n. 01/12748-2) and CNPq (Conselho Nacional de Desenvolvimento Científico – grant. 521264/93-6). The authors also greatly acknowledge the outstanding work of the following students, which implemented great part of the source code: Phd students Mario L.L. Reiss and Roberto S. Ruy; Msc Student Elaine R. Costa.

References

[1] Coelho, L. C. T.; Brito, J. L. N. S., 2002. The E-Foto Project: An Educational Digital Photogrammetric Workstation. In: International Archives of Photogrammetry and Remote Sensing, v. XXXIV, Commission VI, São José dos Campos.
[2] Hasegawa, J.K., 2000. Mapeamento Fotogramétrico com Sensores Digitais. Projeto Fapesp-98/10062-1.
[3] Imai, N.N. et Al., 1999. Recursos não convencionais de levantamento de dados da superfície para aquisição de informações cartográficas. Projeto Fapesp-97/10956-0.
[4] Meneguette Jr. , M.; Meneguette, A. A. C., 1997. Geração de um banco de imagens e mapas digitais. In: Coletânea de Trabalhos- Seminário: Impacto das novas tecnologias na Engenharia Cartográfica. Presidente Prudente.
[5] Pressman, R. S., 1988. Software Engineering, a practitioner's approach. McGraw-Hill.
[6] Siek, J. G.; Lumsdaine, A., 1999. The Matrix Template Library: Generic Components for High-Performance Scientific Computing. COMPUTING IN SCIENCE & ENGINEERING. The IEEE Computer Society.
[7] Tommaselli, A.M.G., 1999. Extração automática de feições lineares em imagens digitais para aplicações cartográficas. Relatório Científico, CNPq, Bolsa PQ.
[8] Tommaselli, A.M.G., 1999. Implantação de recursos de Fotogrametria Digital na FCT- Projeto Fapesp-97/10749-4.
[9] Wis, M., Colomina, I., Gómez, A., Andrinal, P., Miguelsanz, P., 2003, SEIRA: sistema de explotación de imágenes, rápido y aerotransportado, Congreso 5ª Semana Geomática. Barcelona.

References to web pages

[10] DOXYGEN, 2003 http://www.stack.nl/~dimitri/doxygen/
[11] FREEIMAGE, 2003 http://sourceforge.net/projects/freeimage
[12] GANDALF, 2003 http://sourceforge.net/projects/gandalf-library/
[13] GLT, 2003 http://www.nigels.com/glt/index.html
[14] GTK 2003 http://www.gtk.org/
[15] IUE, 2003 http://www.aai.com/AAI/IUE/IUE.html
[16] LIMP, 2003 http://www.remotesensing.org/docs/limp-ht/
[17] NETPBM, 2004 http://netpbm.sourceforge.net/
[18] PVT. 2003 http://www.cv.iit.nrc.ca/~gerhard/PVT/description.html
[19] QT, 2003 http://doc.trolltech.com/3.1/editions.html
[20] TERRALIB, 2003 http://terralib.dpi.inpe.br
[21] TINA, 2003 http://www.niac.man.ac.uk/Tina/tina_intro.html
[22] UCLCV, 2003 http://www.cpsc.ucalgary.ca/Redirect/vision/chain.c
[23] VXL, 2003 http://vxl.sourceforge.net/