TOWARD A GUI REMOTE-SENSING ENVIRONMENT BUILT OVER OTB

David Dubois[1]*, Richard Lepage*[1] *and Tullio Tanzi*[2]

[1] École de Technologie Supérieure, Montréal (Québec), Canada
[2] TELECOM ParisTech, Paris, France

1. INTRODUCTION

With a strong generic programming basis and the use of smart pointers and object libraries, the Orfeo Toolboxe (OTB) provides the remote-sensing community with an impressive set of satellite image processing and classification modules. For a moderately skilled programmer, the programming concepts used greatly ease the modification or addition of modules. Unfortunately, not all remote-sensing researchers are familiar with these concepts. We propose to add a Graphical User Interface (GUI) overlay to OTB in order to open it to a broader community. Using the Fast Light Toolkit (FLTK), this paper shows how the open-source Orfeo Toolbox can be enhanced to take advantage of graphical user interface input for frequently used commands. The command line argument parser is modified to use a graphical user interface to input parameters and images toward various applications built within OTB.

2. PROGRAMMING CONCEPTS

OTB is developed using generic programming concepts [1][2]. This means it offers both great strength against access violations and unparalleled flexibility [3]. This is object-oriented programming taken to its limits. The main reason why OTB was so easily built using these concepts is that it was developed upon the Insight Toolkit (ITK)[1]. OTB also makes use of FLTK for graphical interfaces and GDAL for remote-sensing data importation. The result is a highly modular and powerful toolkit. This makes it easy to modify and build upon what already exists without having to modify a whole lot of code.

3. COMMAND LINE ARGUMENT PARSER

Actually, OTB offers two possibilities for the validation of arguments passed to an application built with this toolkit: leaving it entirely to the programmer or using the generic command line argument parser [4]. The second option is probably the easiest and most reliable to use. This option insures that the programmer defines in detail the various parameters needed for his application and output to user suggested default values as well as descriptive information about each of the parameters.

4. GRAPHICAL ARGUMENT PARSER

The innovation this paper proposes is to use the current methodology as a basis for a dynamically built graphical user interface that will display the list of needed parameters (with description and default values) and save what the user input in order to process it in the same way the original parser does. This means that code modification will be kept to a minimum. Even the new code will be quite short by using object lists to dynamically create the graphical interface according to the number and type of parameters. The intended benefice is that people with low or no programming knowledge of command line manipulations may experiment and use remote-sensing tools developed with OTB. In order to demonstrate the proposed graphical user interface, we slightly modify one tutorial from the OTB User Guide to illustrate the new coding of the graphical argument parser.

[1] http://www.itk.org

5. CONCLUSION

In this paper, we present a way to visually input the needed parameters for applications built with OTB. By extracting the maximum from generic programming and FLTK, it is shown that OTB can be transformed into a graphical remote-sensing toolbox suiting the needs of a broader research community. Future projects include adapting other OTB modules to suit dynamic graphical user interface and eventually provide a visual programming language to graphically create remote-sensing applications [5][6].

6. REFERENCES

[1] Altenkirch, T., C. McBride, et al., “Generic programming with dependent types”, *Datatype-Generic Programming 2006*, Springer-Verlag, Berlin, Germany, pp. 209-257, 2007.

[2] Cirak, F. and J. C. Cummings, “Generic programming techniques for parallelizing and extending procedural finite element programs”, *Engineering with Computers*, Springer-Verlag, Artington, United Kingdom, pp. 1-16, 2008.

[3] Hinze, R. and A. Loh, “Generic Programming, Now!”, *Datatype-Generic Programming 2006*, Springer-Verlag, Berlin, Germany, pp. 150-208, 2008.

[4] OTB Development Team, *The Orfeo Tool Box Software Guide Seventh Edition Updated for OTB-2.6*, Centre National d’Études Spatiales, France, 2008.

[5] I. Bitter, R. Van Uitert, I. Wolf, L. Ibanez, and J. M. Kuhnigk, "Comparison of Four Freely Available Frameworks for Image Processing and Visualization That Use ITK," *IEEE Transactions on Visualization and Computer Graphics,* vol. 13, no. 3, pp. 483-493, 2007.

[6] H. D. K. Le, R. Li, and S. Ourselin, "Towards a Visual Programming Environment Based on ITK for Medical Image Analysis," in *Proceedings of Digital Image Computing: Technqiues and Applications, 2005 (DICTA ' 05)*, pp. 558-565.