OpenStereo

Converting Satellite Image Pairs into Anaglyph Stereoscopic Views

Severino Gomes Neto, Veronica Teichrieb, João Marcelo Teixeira, Judith Kelner
Computer Science Center, Virtual Reality and Multimedia Research Group
Federal University of Pernambuco
Recife, Pernambuco, Brazil
{spgn, vt, jmxnt, jk}@cin.ufpe.br

***Abstract*—Recent years have promoted important evolution in remote sensing, specially taking into consideration quality improvements on satellites that are now capable of obtaining more detailed and accurate observation and producing stereoscopic views. Following market laws, once the level of detail increases the associated costs also rise. In order to take advantage of modest dataset acquisition costs the OpenStereo library (a free, open source, multiplatform, flexible and robust library for stereoscopy) is applied with the purpose of converting satellite images and similar types of geodata acquirers into stereoscopic views.**

Stereoscopic view; depth perception; anaglyph techniques; satellite images

I. INTRODUCTION

An important evolution in the area of remote sensing concerns the massive use of satellite images. Since satellites are able to perform operations like photographing and bringing stereoscopic terrain views, the accuracy of studies around land and oceans could be approached in a more sophisticated manner. Nowadays, the generation of Digital Elevation Models (DEMs) and Digital Terrain Models (DTMs), as well as shaded depth maps is simplified due to the high resolution, precision and accessibility of satellite images.

However, there is a remaining problem related to the acquisition of these stereo images by the community. Someone who desires to acquire a small dataset may usually pay a high price to obtain it, and the cost varies according to the requested data levels of detail. There are several groups of interested people that are not able to afford it, and also the ones that have datasets with a modest level of detail requiring stereoscopic information.

This work aims to make available a non-expensive solution to recycle and improve the information obtained from maps, orthophotos and satellite images, converting their plane aspect into stereoscopic views. The OpenStereo library, which is a free, open source, multiplatform, flexible and robust library for conversion of image pairs into stereoscopic views, was developed and used to perform such conversion [1].

Anaglyph stereoscopic vision is available in some Geographic Information Systems (GISs), but its use implies in costs due to the required acquisition of such a system. Beyond, many users need more detailed information provided by the stereoscopic view for performing specific tasks, such as an urbanism project, but do not necessarily need functionalities provided by a GIS.

In the remainder of this paper some related works are presented in Section II. Section III describes the library used to obtain the stereoscopic views, followed by the presentation of some results. Section V draws a conclusion about the work presented and details some future works.

II. RELATED WORK

The work of Mullard [2] is a similar effort in terms of developing multiplatform stereoscopic solutions. It brings an interesting state of art research and the description of the experience developing a tool capable of viewing stereo pairs in a variety of formats, including anaglyph ones.

Zhang and Xie presented a work regarding satellite image fusion via a web based applet, generating stereo views [3]. The main difference of this work compared to OpenStereo is that it is restricted to a specific application domain related to satellite image anaglyph 3D visualization On the contrary; OpenStereo may be applied to diverse domains.

Commercial solutions are also available, but beyond cost restrictions, they may have an inappropriate and/or hard to use/inflexible interface. Some related commercial products are [4], [5] and [6].

III. THE OPENSTEREO LIBRARY

OpenStereo basically works receiving a pair of images and applying an anaglyph method (true, gray, color, half-color and optimized techniques are available in the current version) selected by the user [7]. Then, it computes this information creating left and right filters that are merged into a single stereoscopic view.

Its architecture is based on the use of plug-ins, the code was written in C++ and is easy to understand and modify. The access to graphics cards capabilities is already available and makes possible a more effective use of the computer power and a higher performance during conversion.

There are some advantages of using OpenStereo, being the most important one the possibility of buying new image

1-4244-1212-9/07/$25.00 ©2007 IEEE.

datasets with fewer details (and consequently lower price) and obtaining stereo views using the library presented here. The same is true for cases when a dataset is already available. In this case, it is not necessary to acquire an expensive newer one only to take more information about the same region.

In order to describe the OpenStereo in more detail, its architecture and the use of plug-ins are discussed further. Beyond, the library's source code is available at the SourceForge.net site [8].

A. *Architecture*

With the purpose of providing an easy manner to integrate existent 3D-based applications to OpenStereo, its architecture is separated in two Application Programming Interfaces (APIs): the first one for converting 3D applications into stereoscopic ones and the second one for supporting the development of plug-ins. The OpenStereo architecture is depicted in Fig. 1.

The first API works providing a set of functions that can be called by a 3D application, allowing to create a viewport (an area for rendering scenes, i.e. a 2D area), to manipulate a camera and to choose the desired anaglyph method.

The second API provides a set of functions that must be implemented by a new plug-in to become usable in OpenStereo's context. It is important to notice that in order to implement a new plug-in developer needs to define the graphics library (OpenGL or DirectX) and the stereo output format to be supported. On the other hand, existing plug-ins can be used without any knowledge about the graphics library or stereo output format supported. Each plug-in is responsible for allocating and rendering stereoscopic data.

The library has three manager modules: `camera manager`, `viewport manager` and `plug-in manager`. The `viewport manager`, which provides the user with possible modifications in viewport size and anaglyph method selection, is responsible for reading the values from the `camera manager` module and allocating the stereoscopic data according to their size. The `Viewport manager` also interacts with the `plug-in manager` module with the purpose of rendering the stereoscopic scene.

OpenStereo loads all plug-ins and accesses the entry points through `plug-in manager` functions, requiring two steps: creation of a context and OpenStereo's initialization. Creating a context consists in configuring memory to allocate viewports, plug-ins, and cameras, and controlling them through internal OpenStereo data structures. Initializing OpenStereo means loading a file containing information about the available plug-ins and the definition of a standard one used to generate the stereoscopic output.

Since the steps mentioned are completed, the plug-ins must be loaded and their entry points (symbols) should be accessed. In case any symbol is missing, an error occurs to indicate the plug-in can not be used. The symbols represent seven minimal operations needed to allow the use of a plug-in, as follows:

1. Plug-in loading – it allows plug-ins to be checked and loaded, setting the present variables;

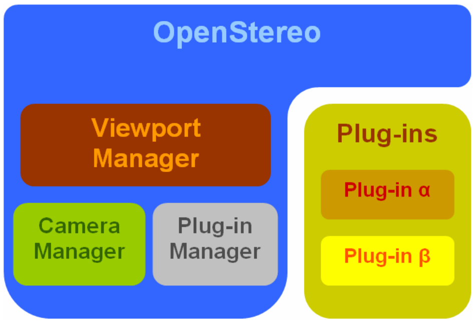

Figure 1. OpenStereo architecture.

2. Name querying – it provides the application a way to check the plug-in identification;
3. Description querying – it provides access to the detailed plug-in information;
4. Stereo data allocation – it checks and, if possible, sets apart an amount of memory capable of storing the stereo data;
5. Stereo data destroying – it collects the memory used in the prior operation since it is not required anymore (it is often used when viewport changes its size);
6. Stereo data rendering – it performs the stereo data rendering in the viewport considering the graphics library in use;
7. Plug-in unloading – it collects the variables allocated by operation 1 and allows a plug-in to be "unplugged".

Details about the use of plug-ins and the explanation of their action flow are presented in next subsection.

B. *Plug-ins*

An application that uses OpenStereo and a valid plug-in needs to perform the following steps in order to generate stereoscopic content accordingly: create a context, load a plug-in, create a viewport and a camera, associate the camera to the viewport (changing viewport size when it is needed) and link the viewport to the context to finally position the camera and render the scene. This detailed flow is shown in Fig. 2.

Camera can be posed using the toe-in method, which has a satisfactory visual result but introduces a vertical parallax as depicted in Fig. 3., or the off-axis method, shown in Fig. 4. This last method corrects this problem, but is more difficult to implement.

In order to convert satellite images, a simple application was developed creating a viewport with the same dimension of the pair of images, assuming the disparity between them as different camera poses and rendering the scene in a new stereoscopic image.

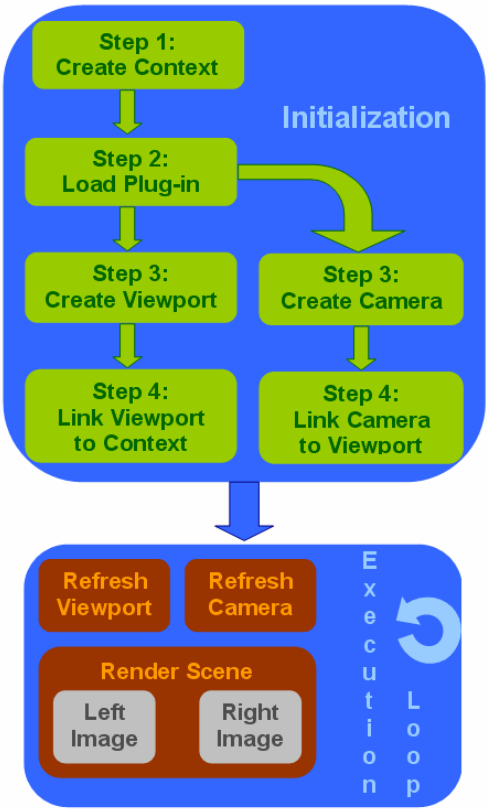

Figure 2. Execution flow in OpenStereo.

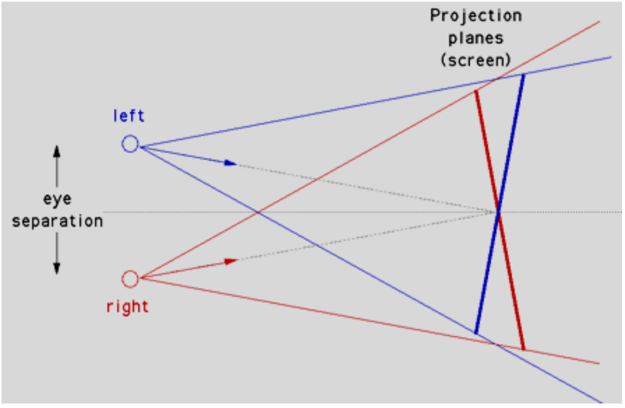

Figure 3. Toe-in method.

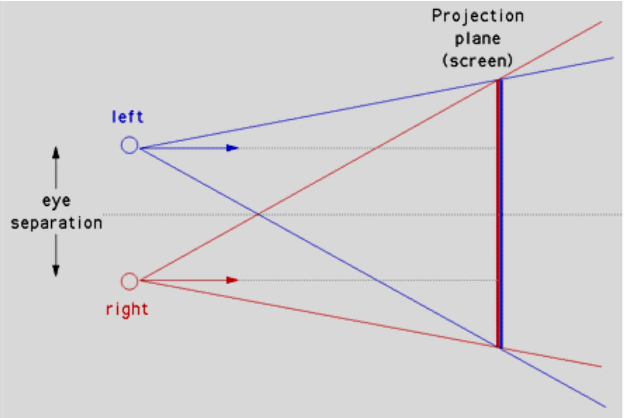

Figure 4. Off-axis method.

IV. RESULTS

In the experiments using narrow disparity the results were realistic enough with the use of simple anaglyph glasses. On the other hand, if the pairs of images have wide disparities the result could be unsatisfactory. This is expected because of the dependency of proximal points of view for generating a more accurate depth perception.

Fig. 5. and Fig. 6. demonstrate the level of accuracy and detail provided when using the solution with small disparity.

V. CONCLUSION AND FUTURE WORK

Since OpenStereo could be satisfactory evaluated in this work, the authors intend to analyze how viable it is using the library in other studies in the vast area of geosciences. Other planned effort is to integrate this solution in DEM/DTM free or open source applications.

The OpenStereo library is technically easy to understand, use and modify with results as accurate and detailed as the pairs of images used and, monetarily, it is a free and accessible library. Because of these reasons it can be noticed the importance of OpenStereo's use on satellite images stereoscopic viewing and its promising possibilities in geoscience and remote sensing areas.

REFERENCES

[1] P. J. S. Leite, J. Kelner and V. Teichrieb, OpenStereo: uma biblioteca para suporte ao desenvolvimento de aplicações estereoscópicas. Recife, Brazil: Federal University of Pernambuco, BSc Final Monograph, 2006.

[2] R. Mullard and N. Holliman, Distributed stereoscopic image viewing. Durham, United Kingdom: Durham University, M.Sc. Thesis, 2005.

[3] Y. Zhang, and P. Xie, "Automatic image fusion and web-based automatic colour 3D visualization," American Society of Photogrammetry and Remote Sensing Annual Conference, May 2004.

[4] PCI Geomatics' FLY! Software. Available in: PCI Geomatics. URL: http://www.pcigeomatics.com/products/fly.html, visited on April, 2007.

[5] Stereo Analyst. Available in: Leica Geosystems. URL: http://gi.leica-geosystems.com/LGISub1x30x0.aspx, visited on April, 2007.

[6] Create Anaglyphs Using ArcScene. Available in: ESRI. URL: http://www.esri.com/news/arcuser/0704/files/anaglyphs.pdf, visited on April, 2007.

[7] P. Wimmer, Anaglyph Methods Comparison. Available in: Peter Wimmer Homepage. URL: http://mitglied.lycos.de/stereo3d/anaglyphcomparison.htm, visited on April, 2007.

[8] SourceForge.net: OpenStereo. Available in: SourceForge.Net. URL: http://sourceforge.net/projects/openstereo/, visited on April, 2007.

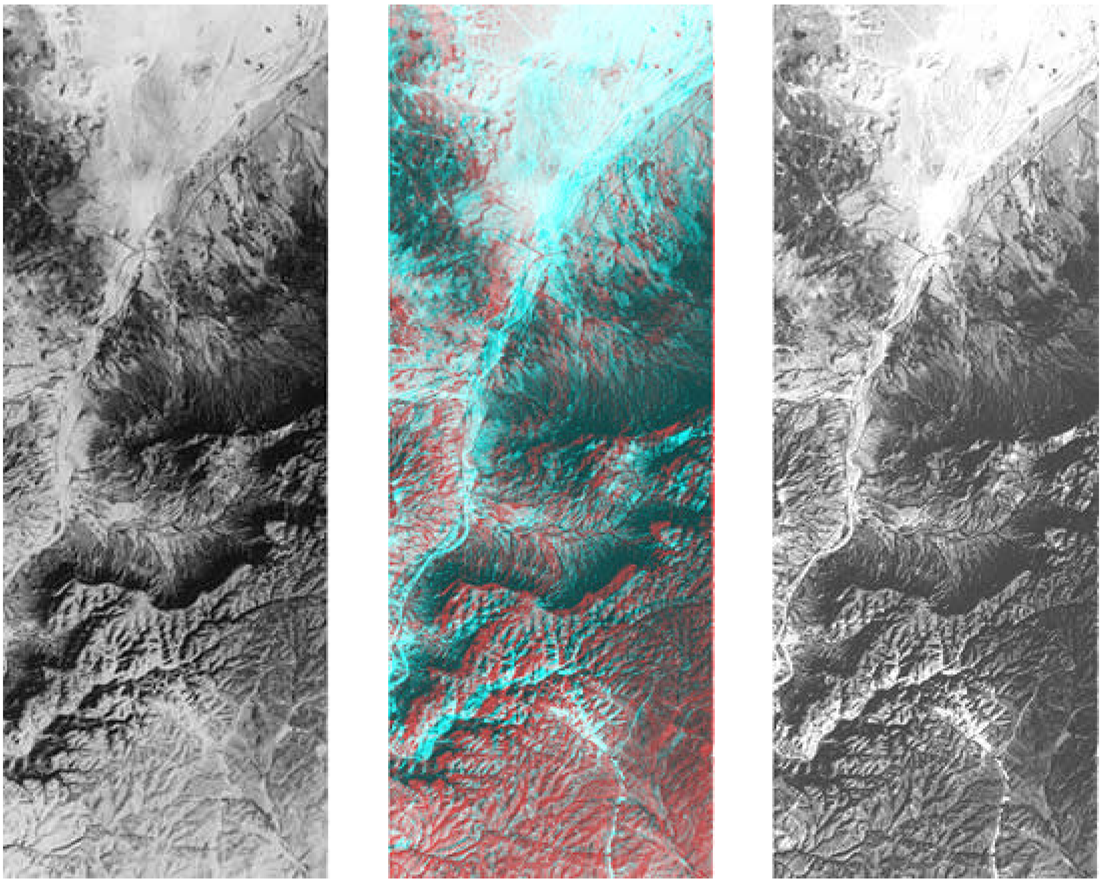

Figure 5. Left and right images are combined into a resultant stereoscopic view at the center.

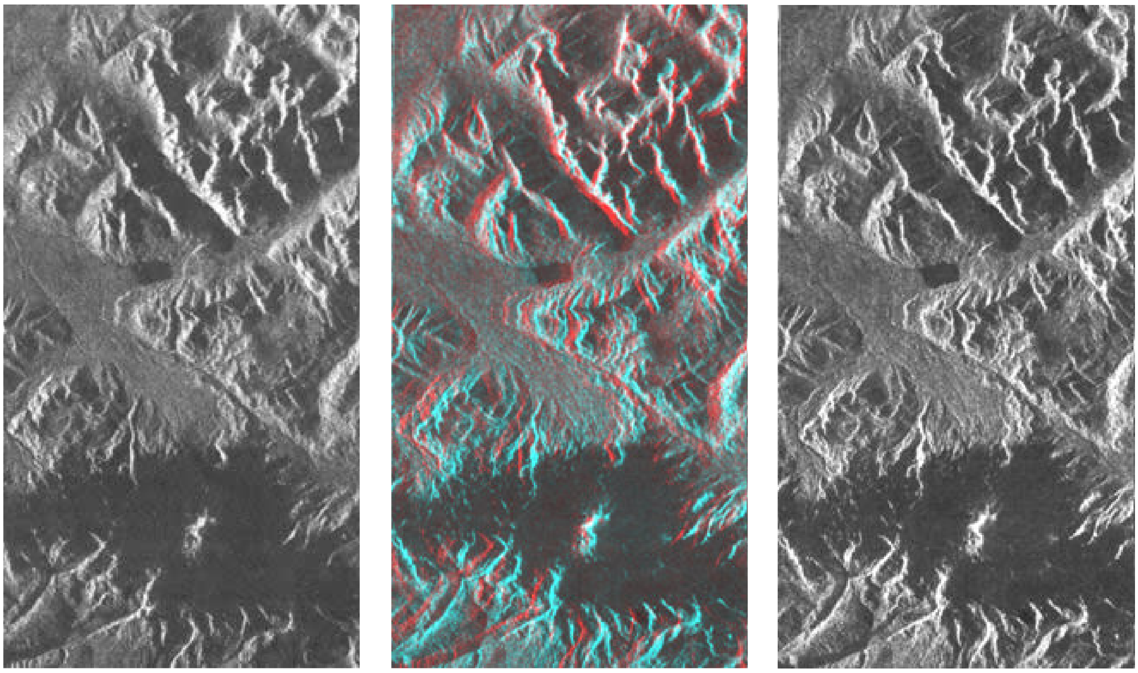

Figure 6. In the stereoscopic image at the center it is possible to distinguish the height variation of the terrain.