Title: Woolz IIP: A Tiled On-the-fly Sectioning Server for 3D Volumetric Atlases

Authors: Zsolt L. Husz, Thomas P. Perry, Bill Hill, Richard A. Baldock

Book: Advances in Visual Computing

Book Series: Lecture Notes in Computer Science

Proceedings: 5th International Symposium, ISVC 2009

Editors: G. Bebis et al.

Volume: 5875/2009

Date of publication: 26/11/2009

Pages: 924–933

Publisher: Springer Berlin / Heidelberg

DOI: http://dx.doi.org/10.1007/978-3-642-10331-5_86

ISBN: 978-3-642-10330-8

Project page: http://www.emouseatlas.org/Software/WlzIIP

Funding: MRC, UK and NIH #1R01MH070370-01A2

MRC Human Genetics Unit

Woolz IIP: A Tiled On-the-fly Sectioning Server for 3D Volumetric Atlases*

Zsolt L. Husz[1], Thomas P. Perry[2], Bill Hill[1], Richard A. Baldock[1]

[1] MRC Human Genetics Unit, Institute of Genetic and Molecular Medicine, Western General Hospital, Edinburgh EH4 2XU, UK
{Zsolt.Husz, Bill.Hill, Richard.Baldock}@hgu.mrc.ac.uk
[2] Institute for System Level Integration, Alba Centre, Alba Campus, Livingston, EH54 7EG, UK,
Thomas.Perry@sli-institute.ac.uk

Abstract. We present a novel method to provide fast access to large 3D volumetric data sets from biological or medical imaging atlases. We extend the Internet Imaging Protocol with an open specification for requesting tiled sections of 3D objects. We evaluate the performance of the protocol and demonstrate it with a platform independent web viewer that allows on-the-fly browsing of section views of multi-gigabyte 3D objects.

The method uses *Woolz*, an efficient image processing library, to provide very fast access to section views of the volumetric data. The server has been implemented to run on standard Linux systems and it avoids the requirement for high-performance parallel processing or expensive software. We have tested the system on data volumes up to 13.4 GB and demonstrated no loss of responsiveness for the user.

1 Introduction

In the field of biomedical science, the ability to access 3D image objects over a wide-area network such as the internet is often imperative. Previous solutions involve the Internet Imaging Protocol (IIP) [1], which is an open protocol that provides fast tiled delivery of large images through a multi-resolution image representation, but which may only be used with 2D images. Since a similar presentation method for 3D objects does not exist, we have developed extensions to IIP which we call *Woolz Internet Imaging Protocol* (WlzIIP), implemented a server to provide this service and built a web application to use it.

Recent work has demonstrated the clear advantages of tile-based image transmission and many *zoom-viewers* have been developed for example by Google and Zoomify. The IIP server has been developed as an open-source resource and is used in tele-pathology and educational archives [2]. It allows a user to select a region of interest at a desired zoom level and provides efficient image transmission.

The importance of virtual slicing systems for remote access of images was previously noted [3] and the IIP protocol was identified as a suitable interface

* We acknowledge support from the Medical Research Council, UK and NIH support under grant #1R01MH070370-01A2.

This paper was published in Advances in Visual Computing, 5th International Symposium, ISVC 2009, Springer Berlin / Heidelberg. The original publication is available at www.springerlink.com, http://dx.doi.org/10.1007/978-3-642-10331-5_86

for independent client-server applications. However the availability and flexibility of these imaging systems was limited by the proprietary (and costly) nature of existing implementations. Some existing image servers are able to deliver 3D image data, such as in BrainMaps.org, they provide only predefined 2D sections. Glatz-Krieger et al. [3] consider virtual slices only in the original focal planes of the biological material in the context of a 2D microscope slide. In this paper, we cut an *arbitrary* virtual section from the digitised 3D model. Such sectioning software exists, either as standalone (e.g. Amira [4]) or as online Java applications (e.g. NeuroTerrain [5]). The latter aims for compatibility and platform independence, but sometimes falls short of this in practice: for example, Iowa Slidebox [6] suffers from its binding to an obsolete Java runtime environment.

Tile based image delivery, that transmits the target as smaller image blocks, is known from commercial web applications such as Google maps. This runs in any web browser and does not require additional software or an applet.

The Visible Human project has generated several internet based image servers and clients. The EPFL server [7] is the most similar to ours. It is a high throughput parallelised sectioning server using a FastCGI (FCGI) web interface. However, compared to WlzIIP, it does not allow tiled requests and has a proprietary protocol. To deliver section data the EPFL server requires a high performance cluster. In contrast, the WlzIIP server will run on standard Linux-based servers with the only requirement that the installed memory is larger than the image volume.

Sections 2 and 3 present the internal image representation and 3D sectioning that provide fast image generation. Sect. 4 explains our WlzIIP extension of IIP. Then, a visualisation interface using the WlzIIP is presented in Sect. 5, and in Sect. 6 we compare WlzIIP to the NeuroTerrain image server. The paper ends with a discussion and conclusions.

2 3D object representation

As part of our work, we use the image processing library known as *Woolz* [8]. Internally *Woolz* uses an interval coded representation for objects which is efficient with regard to both sparse data storage and image processing operations.

In 2D, an image is defined over an arbitrary region of a discrete 2D space with coordinates (k, l) where k is the column coordinate and l the line coordinate. For each line in the image there is a list of intervals which gives the start and end points of the image along that line. There is a list (possibly empty) of intervals for each line and it is clear that an arbitrarily complex region of the discrete space can be defined in this way. It is assumed that the discretisation in the x and y directions is at fixed regular intervals, constant in both directions but not necessarily equal. The 3D structure is simply a stack of 2D images. The plane coordinate is defined to be p, where the planes are evenly spaced, each with a 2D image, or possibly an empty structure.

The advantage of the Woolz encoding is that only grey-level information within the domain of the image is stored rather than for the whole rectangular box defined by the column, row and plane bounds. For a biological atlas this reduces storage and memory requirements with lossless compression.

3D object reconstructions are built with specialised Woolz tools for section data registration. In this paper the reconstruction technique is immaterial and not discussed and in fact the WlzIIP can be applied to any volumetric image.

Each object has its own internal discrete coordinate system with an associated affine transform which will provide the link between internal coordinates and external, biologically relevant coordinates.

2.1 Coordinate Transformation

There are many ways to define an arbitrary rotation, scaling and translation of one coordinate frame into another. For the purposes of sectioning we use a set of parameters that are chosen to correspond to those used in the MAPaint Woolz viewing tool to select arbitrary planes through reconstructions. The underlying coordinate transformation methods have been extensively used for developing the e-MouseAtlas models and gene-expression database.

We define a viewing plane with a new set of coordinate axes such that the new z-axis is along the *line-of-sight*. The viewing plane is defined to be perpendicular to the viewing direction given by angles θ and ϕ which are yaw and pitch respectively. The actual plane is distance d from the fixed point $\mathbf{f}$. Internally the full rotation transformation is defined in terms of the Euler angles [9, p. 107] with British definition [10, p. 9]. The third degree of freedom (d.o.f.) is called *roll* and for the user corresponds to rotating the section image as viewed on the screen. In many cases the user will want a *standard* view of the data without the requirement of an additional control to set the viewing angle, so we have implemented a number of viewing modes which automatically determine this angle.

3 Viewing modes

With views that are perpendicular to the line of sight, we present four options to determine the orientation of the section image on the screen.

Statue mode: the viewing plane is *flat* but the image is oriented as if the viewer were *walking around* the object. The actual displayed image is then obtained by rotating the viewed plane about an axis parallel to the line of intersection of the view plane and the *horizontal* which is defined to be a plane of constant z. This has the merit of providing clear feedback of the position of the plane within the whole but is not ideal because for some angles the projection will introduce perspective distortion of the image.

Up-is-Up mode: the projection of a predefined direction *up* will always be displayed as the vertical in the section view. If the viewing direction is parallel to this vector then the angle of rotation around the viewing direction is not defined and an arbitrary choice can be made. As a consequence, small changes in viewing direction around the *up* vector may give rise to arbitrarily large changes in the display orientation.

Fixed point mode: navigation through a 3D volume can give rise to confusion if unfamiliar views are presented. However it may often be possible to identify one or more points within the image volume that the user wishes to be visible. If one point is fixed then there are two d.o.f. left to set the view and if there are two fixed points then there is only one d.o.f.

The transformation is defined so that by setting one fixed point, $\mathbf{f}$, the orientation parameters, θ and ϕ, will rotate the view plane about this point.

Fixed line mode: if two points are fixed then θ and ϕ are dependent and can be represented in parametric form using a third angle parameter, ψ, which corresponds to the angle around the line joining the two fixed points.

The two fixed points $\mathbf{f_1}$ and $\mathbf{f_2}$ give direction vector $\mathbf{n_1} = \frac{\mathbf{f_2}-\mathbf{f_1}}{|\mathbf{f_2}-\mathbf{f_1}|}$ which must remain in the view plane. The values of pitch and yaw of the original plane in which the fixed line was established define a direction perpendicular to this vector $\mathbf{n_1}$ and can be used to establish the formula linking ψ to new viewing angles.

This technique has proved very powerful and is widely used in MAPaint.

4 Tile based imaging with WlzIIP

In this section we present the extension of WlzIIP to the IIP protocol.

4.1 Image tiles

In WlzIIP, we keep the tile based imaging capability of IIP and so each image is divided into fixed sized tiles (except right-most or bottom-most tiles which might be smaller).

Displaying 3D objects involves multiple coordinate systems, and WlzIIP will automatically perform transformations between these. For clarity, the conventions used are explained in Fig. 1.

4.2 Protocol extension

The added commands and feature queries are shown in Tables 1 and 2.

The commands specify an object, set the viewing section parameters and request image data or metadata, similar to existing IIP parameters. The commands for image requests are the same as in the original IIP specification [1]: **CVT** for full frame; **JTL** and **TIL** for jpeg-compressed and uncompressed tile answers. For a Woolz object, **SCL** specifies an arbitrary scaling factor, so resolution number is ignored in **JTL** or **TIL** commands.

For 2D images, pyramidal tiled TIFF images are specified by the **FIF** command, while for WlzIIP the **WLZ** command sets the 3D Woolz object. These are cached in the server's memory for efficiency.

MOD specifies the projection mode being `STATUE`, `UP_IS_UP`, `FIXED_LINE` or `ZETA` (fixed point) as described in section 3.

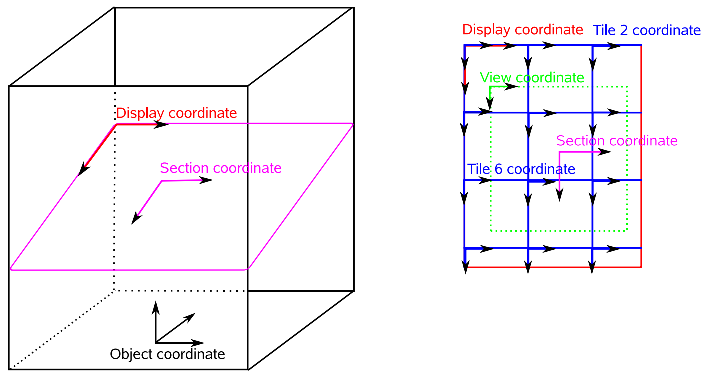

Fig. 1. Coordinate systems. The object coordinate is defined by the object, and the section coordinates result from the sectioning process. The origin of the sectioning coordinates is in an arbitrary position with respect to the visible pixels of the sectioning plane. However, images are normally represented using pixels with positive coordinates. Hence, the display coordinates translate section coordinates such that the lower bound of the bounding box of the visible pixels is at $(0, 0)$. Further, on the right, the section is divided into non-overlapping tiles covering the whole section. Tiles are numbered with 0, 1, 2, etc., with the coordinates of the 0^{th} tile (i.e. top-left corner) matching the display coordinates. The 1^{st}, 2^{nd}, etc. tiles continue from left to right and top to bottom in raster fashion. Finally, the view coordinates are defined for the region of the reassembled tiles displayed in the viewer application.

Table 1. WlzIIP command extension summary

Command	Purpose	Syntax
WLZ	Specify the Woolz object	`WLZ=`*`path`*
MOD	Specify the projection mode	`MOD=`*`mode`*
DST	Specify the distance of the sectioning plane	`DST=`*`dis`*
PIT	Specify the pitch angle of the sectioning rotation	`PIT=`*`angle`*
ROL	Specify the roll angle of the sectioning rotation	`ROL=`*`angle`*
YAW	Specify the yaw angle of the sectioning rotation	`YAW=`*`angle`*
SCL	Specify the scale used in the sectioning transformation	`SCL=`*`scale`*
FXP	Specify the fixed point of the viewing section rotation	`FXP=`*`X,Y,Z`*
FXT	Specify the second fixed point of the viewing section rotation	`FXT=`*`X,Y,Z`*
PAB	Specify the 3D query point absolute in the object coordinate	`PAB=`*`X,Y,Z`*
PRL	Specify the 2D query point relative in tile or display or tile coordinate	`PRL=`*`T,X,Y`*
UPV	Specify the up vector for the up is up mode	`UPV=`*`X,Y,Z`*

DST sets the viewing plane distance, while **PIT**, **ROL** and **YAW** set the plane angles. Other commands set parameters specific to the viewing mode.

The retrieval of a tile is an HTTP request that includes a combination of the above commands. Such an example is shown in Fig. 2.

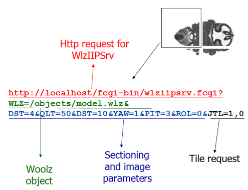

Fig. 2. WlzIIP tile request example. The HTTP request consists of the web address of the FCGI server, the specification of the 3D Woolz object, the sectioning and image parameters and finally the tile request command. The response is the first tile out of the four of a 3D tomographic object section.

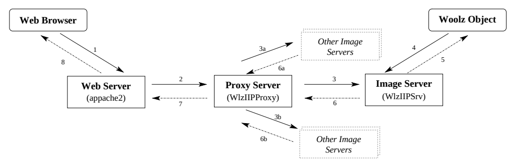

Fig. 3. Architecture of WlzIIP server using a proxy server. The web server passes the user requests to the proxy, which forwards them to individual IIP servers. These servers have direct access to the Woolz Object and return the requested data. The numbered lines show the order of the requests (continuous lines) and the replies (dotted lines).

Table 2. WlzIIP object request extension summary

Object	Purpose
IIP-server	Identify if WlzIIP server is running
Max-size	The size of the section
Tile-size	The size of a tile
Wlz-true-voxel-size	The voxel size of the object
Wlz-volume	The volume of the object
Wlz-distance-range	The range of the sectioning plane distance
Wlz-sectioning-angles	The pitch, yaw and roll angles of of the sectioning plane
Wlz-3d-bounding-box	The first and last plane, line and column number of the object
Wlz-coordinate-3D	The 3D coordinates defined in 2D by the **PRL** command
Wlz-grey-value	The grey or RGB value of a point specified either the **PRL** or the **PAB** commands

Features of a given section can be obtained with **OBJ** queries listed in Table 2. We provide also coordinate translations and voxel queries. An example query for a sectioning plane distance range is

```
http://localhost/fcgi-bin/iipsrv.fcgi?YAW=61&PIT=3&ROL=0&MOD=ZETA
&WLZ=/objects/small.wlz&OBJ=Wlz-distance-range
```

which results in the reply

```
Wlz-distance-range:0 171
```

Our C++ software is based on a GPL implementation of the IIP server by Pillay and Pitzalis [11]. This is a FastCGI (FCGI) web server module that is called by the web server (e.g. Apache).

4.3 WlzIIP Proxy

To handle multiple requests, large objects and to provide a single access point to image servers separated from the Internet by a firewall, we have developed a tool called WlzIIPProxy that filters FCGI requests and forwards them to different WlzIIP servers. The communication conforms to the FCGI protocol. Though it was designated to work for IIP and Woolz requests, it is generic and can route any FCGI request, hence it is also possible to chain multiple proxies.

The multiple WlzIIP server architecture is shown in Fig. 3. WlzIIPProxy is an independent program running on the proxy server. The web server (e.g. Apache 2) forwards the FCGI request to this server on a configurable port, then the HTML request string is checked by WlzIIPProxy and if the definition string of any remote WlzIIP server is a substring of the request parameters then this query is forwarded to the matching server. If no correspondence was found then the request is passed to the default server.

5 Web viewer prototype with WlzIIP

For testing WlzIIP, we have developed a JavaScript application that runs in a web browser, based on the viewer of Pillay [12]. The WlzIIP viewer allows browsing through the objects with four controls. These change the pitch and yaw angles of the sectioning plane in the fixed point mode, alter the sectioning plane distance and the zoom level, and pick the current viewing region in a thumbnail view.

The browser requests tiles only for the currently viewed regions at the nominal screen resolution, so no transmission bandwidth is lost for non-visible regions or for unrepresentable details. When a user pans the object, new tiles are requested and displayed. For an $N \times N$ image, scrolling in one direction requires at most $N + 1$ new tiles. By reducing the size of the transmitted data in this way, the transmission throughput is increased by $\frac{N^2}{N+1}$. For $N = 10$, plausible for biological images, this provides a performance increase factor of 9.09.

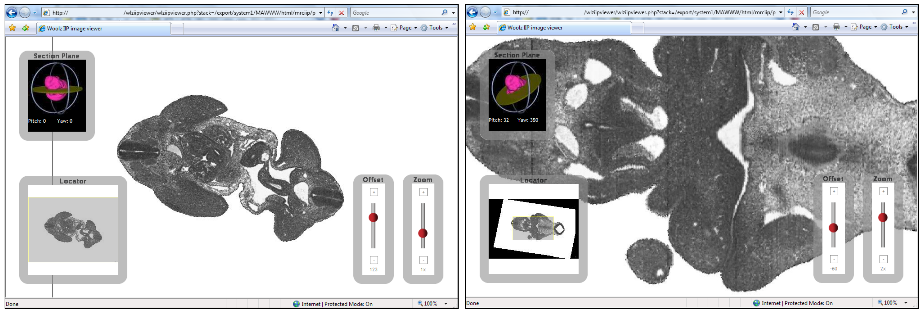

Fig. 4. Web interface using WlzIIP showing two views of a 3D object. The *Section Plane* selects pitch and yaw angles, the *Locator* control provides a thumbnail and allows viewing zone selection, the *Offset* control sets the sectioning plane distance, and the fourth control provides a zoom in the range of 0.25x–4x.

The caching mechanisms of the web browser and of the WlzIIP server reduce the response time for tiles, sections and objects.

Tests were performed from a university network, a home ISP and free low-bandwidth wireless access on a train, and all tests show fast response times and great capability for interaction. Currently our application is compatible with Firefox, Internet Explorer, Safari and Opera browsers on Microsoft Windows, Linux and Mac OS X operating systems.

The WlzIIP project webpage [3] provides demos and further information about both WlzIIP server and viewer.

6 Evaluation

We have evaluated the WlzIIP server and compared it to NeuroTerrain [5]. Our test requested 1056 consecutive sections of a 3D object imported from NeuroTerrain. Each grey level section of the default resolution image consists of 3×5 tiles of 128×128 pixels. The average retrieval time of 10 repeated tests provided the results from the bottom five lines of Table 3.

The table includes NeuroTerrain high and low bandwidth results, full section requests using the **CVT** command and four tiled retrievals on a local client (1 Gbps LAN), on two remote clients on the JANET[4] network (1 Gbps backbone) and on a client with 2 Mbps home broadband.

In NeuroTerrain, the full sized, uncompressed grey image is transmitted, therefore real and browsing frame rates are equal. Also, the pixel throughput equals the data throughput.

First, to compare WlzIIP with NeuroTerrain, full frame requests with **CVT** were tested. This has a throughput 6.4 times lower than the highest throughput

[3] `http://www.EMouseAtlas.org/Software/WlzIIP`

[4] JANET is the UK educational and research network

Table 3. WlzIIP server evaluation. The throughput is the speed of the (compressed) data transmission; the pixel throughput is the pixel transmission rate; the real frame rate is the speed of transmission of full images; the browsing frame rate is the estimated frame rate that users experience.

	Through-put [KB/s]	Pixel through-put [Kpixel/s]	Real frame rate [fps]	Browsing frame rate [fps]
NeuroTerrain [5]				
LAN	4060.00	4060.00	5.47	5.47
DSL	141.00	141.00	0.20	0.20
WlzIIP full (CVT)				
LAN	634.10	6993.91	37.31	37.31
WlzIIP tiled (JTL)				
LAN	107.49	929.61	4.96	45.08
JANET metropolitan	80.26	694.09	3.70	33.66
JANET remote	54.68	472.83	2.52	22.93
Home broadband	17.50	151.31	0.81	7.34

of NeuroTerrain. However, for WlzIIP the pixel transfer rate is higher due to the compressed image data, which results in a higher frame rate.

Tiled **JTL** requests have a lower real frame rate. However, the tiled approach has the advantage of browser caching. Therefore, the browsing frame rate takes into account an increase of 9.09 times (as estimated in Sect.5). Note that the non-zoomed NeuroTerrain frames are too small to benefit from this directly. However, the speed of the magnified image and of larger datasets improves considerably. The browsing frame rate is superior to the full image based transmission even using the home broadband connection, i.e. equivalent to the DSL tests on NeuroTerrain [5].

7 Discussion and conclusions

The main contribution of this paper is the extension of the IIP protocol for 3D objects that allows fast sectional data browsing over the Internet.

Compared to local image and object viewers, the WlzIIP server offers central management of the image content and storage of the object at the provider, thus it allows simple update and deployment of new content. Specification of the details (e.g. zoom level and spatial localisation) of the region the user is interested in permits a reduction in the size of the transmitted information, and hence allows fast interactive access to large data objects.

The main disadvantage of our current WlzIIP server is the requirement of sufficient memory to load 3D the object. However, this drawback is limited: using WlzIIPProxy, large objects can be distributed over multiple servers for which memory has become a low cost resource. Also, up to our largest dataset of 18.5 GB we have not observed performance degradation after the initial object disk read. With the underlying Woolz architecture, the extension to read partial object is straightforward, although currently object sizes don't require this.

Tiled images increase browsing frame rate whilst reducing the size of the transmitted data. Compatibility with HTTP allows portability and simple integration into client applications. Being browser based, the WlzIIP client does not need local installation for viewing nor does it need an additional IP port for communication.

Other web based delivery systems, such as the original IIP specification and Google Maps, are restricted to 2D. However, WlzIIP delivers sections from 3D objects. Similar to IIP, WlzIIP is an open protocol and the server code is freely available from us with a GPL licence.

To summarise, our main achievement was to extend the standard IIP protocol to allow fast sectioning of 3D objects using different sectioning modes and parameters, metadata and section-coordinate queries of the object. This has had the effect of increasing the frame rate experienced by the user. Using WlzIIP-Proxy, the method is scalable. The portable JavaScript browser does not require special software or applets to run locally and provides a highly interactive browsing environment. The WlzIIP server is currently being integrated in biological atlases such as EMAP, EurExpress and EuReGene and we are looking forward to further applications.

References

1. I3A: Internet imaging protocol (1997), version 1.0.5.
2. Mea, V.D., Roberto, V., Beltrami, C.A.: Visualization issues in telepathology: The role of the internet imaging protocol. In: 5th Int'l Conf. on Information Visualization, pp. 717–722 (2001)
3. Glatz-Krieger, K., Glatz, D., Mihatsch, M.J.: Virtual slides: high-quality demand, physical limitations, and affordability. Human Pathology 34(10), pp. 968–974 (2003)
4. Stalling, D., Westerhoff, M., christian Hege, H.: Amira: A highly interactive system for visual data analysis. In: The Visualization Handbook, pp. 749–767, Elsevier (2005)
5. Gustafson, C., Bug, W.J., Nissanov, J.: Neuroterrain – a client-server system for browsing 3D biomedical image data sets. BMC Bioinformatics 2007 8(40), (2007)
6. Heidger Jr., P.M., Dee, F., Consoer, D., Leaven, T., Duncan, J., Kreiter, C.: Integrated approach to teaching and testing in histology with real and virtual imaging. The Anatomical Record Part B: The New Anatomist 269(2), pp. 107–112 (2002)
7. Bessaud, J.C., Hersch, R.D.: The visible human slice sequence animation web server. In: 3rd Visible Human Project Conf. Proc., (2000)
8. Piper, J., Rutovitz, D.: Data structures for image processing in a c language and unix environment. Pattern Recognition Letters 3, pp. 119–129 (1985)
9. Goldstein, H.: Classical Mechanics. 2 edn. Addison-Wesley, Reading, MA (1950)
10. Whittaker, E.T.: A Treatise on the Analytical Dynamics of Particles and Rigid Bodies. 3 edn. Cambridge University Press, London (1927)
11. Pillay, R., Pitzalis, D.: IIPSrv, v. 0.9.7, (1997), http://prdownloads.sourceforge.net/iipimage/iipsrv-0.9.7.tar.bz2
12. Pillay, R.: IIPMooViewer, v. 1.0, (2007), http://iipimage.sourceforge.net