



[osmia rufus]

OSMIA

Deliverable D3.3

TINA Interface System for Voxar3D (Plug n View)

**Prepared by Ian Poole for Voxar Ltd.
July 2003**

Voxar Ltd
Bonnington Bond
2 Anderson Place
Edinburgh EH6 5NP
T : +44 131 472 4792
F : +44 131 472 4799

<http://www.voxar.com>

Overview

We wish to use the TINA libraries (tina-libs) to prototype new functionality in Voxar3D (previously named “Plug n View”).

Issues

- Need to compile under Windows and link with programs built under Visual Studio
- Need to inter-operate with our in-house imaging libraries, requiring conversion between image types.

Failed approach – building via Cygwin

We at first attempted to build the libraries using the ‘configure’ system set up by ISBE, for UNIX-like platforms. As well as using the identical compiler (gcc) this approach has the advantage that add/remove/rename of files by developers on other platforms would be automatically incorporated into the Windows build. This required the build to be performed under the Cygwin environment (www.cygwin.com), which must be installed, along with more recent versions of autoconfig. Having got the build to work under Cygwin we then had to address the problem of linking the library to an application built under Visual Studio. This proved problematic, perhaps because Cygwin was using different versions of the standard header files to Visual Studio. Doubtless these problems could be solved, but they were taking much time so we eventually abandoned this approach.

Direct build in Visual Studio

Instead, we adopted the pragmatic approach of building the libraries within a Visual Studio project. This required us to make some minor changes to the source to accommodate compiler differences, but ensures that subsequent linking goes smoothly. Although Visual Studio has many shortcomings, it is familiar to most Windows developers, and so provides a convenient way of browsing the libraries. Compilation is also substantially faster than via Cygwin/gcc.

In concrete terms, we have added the directory `tina-libs/vs_build` to CVS repository, and also handcrafted a version of `config.h` compatible with the windows platform. We have also added a separate Visual Studio for the simple example program, serving to demonstrate the success of the build.

Interfacing with Voxar libraries

In order to use the TINA libraries at Voxar it was necessary to convert between TINA’s 2D imaging structure – `Imrect`, and the equivalent Voxar structure – `VyImage`. To avoid problems arising from data ownership, we elected to do a full copy of the data in their conversion. `Imrect` and `VyImage` both support byte, short, unsigned short, long, float and double pixel types. However, `Imrect` also has

complex and pointer types, which `VyImage` does not, so in this case an error is raised.

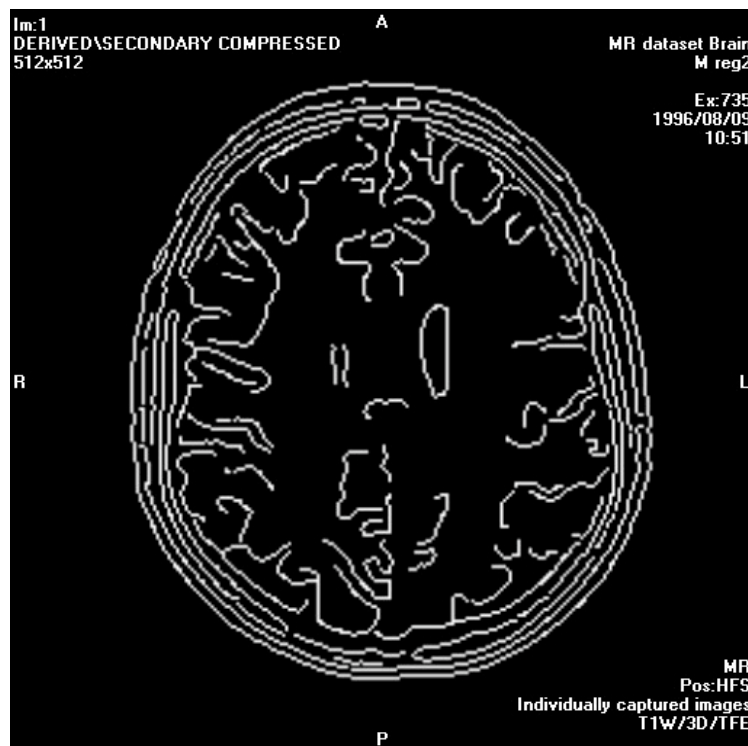
The conversion code consists in static functions inside a class to keep the namespace clean:

```
class VyTinaConversions
/* Class of static functions, to convert between Tina and VyLib types
*/
{
public:
    static Imrect* CopyToTinaImrect (const VyImagePtr& vyim);
    static VyImagePtr CopyToVyImage (Imrect* tinaIm);
    static DataType ToVyLibDataType (Vartype tv);
    static Vartype ToTinaVartype (DataType type);
};
```

Applications

We have linked the TINA libraries, along with the conversion routines described above, into a branched version of Voxar3D aimed at previewing advanced functionality at the November 2003 RSNA exhibition in Chicago. Advances in segmentation are a particular goal, and TINA routines are assisting with this. We are making direct use of TINA routines for matrix manipulation, tangential smoothing, Canny edge detection and MR coil correction.

The image below shows the result of TINA's `canny()` function, displayed in Voxar3D.



We are using the Canny edge detector as part of an implementation of 'power assisted' interactive tracking of edges.

We will take the prototyping to a point where it can be evaluated by our clinical application specialists.