NAME

Vtutorial – Vista tutorial

INTRODUCTION

Vista provides a set of software modules and an environment for doing computer vision research. It currently contains modules for a wide range of image processing and higher-level vision operations. Vista is not limited to operating with images, but also provides a user-extensible set of data types for manipulating vectors and other types of data.

A key aspect of Vista is that it does not just provide many existing modules, but is also designed to make it easy for any user to add their own modules and define new data types. A large library of subroutines is provided to simplify programming tasks. All code is written in ANSI C, and an interface from Common Lisp is provided. Each module can typically be run as a Unix filter or called as a subroutine. In addition, a number of modules provide graphical interfaces for inspecting and manipulating data, and a toolkit is provided to simplify the task of creating interactive programs running under the X Window system.

Vista was developed at the University of British Columbia's Laboratory for Computational Intelligence. The initial developer of the environment was Art Pope, but other members of the laboratory are contributing to its further development. All code is available to the public in source form without significant restrictions, so we hope that it will become a standard for development and interchange in the vision community.

GETTING STARTED

You need to ensure that:

- the search path for commands executed by your shell (the **path** shell variable) includes the directory **/u/cs449/bin**, where Vista programs are installed:

  set path = ($path /u/cs449/bin)

- the search path for on-line manual pages (the **MANPATH** environment variable) includes the directory **/u/cs449/man**, where Vista manual pages are installed

  setenv MANPATH $MANPATH":/u/cs449/man"

- the **VISTA** environment variable is defined as the location of the Vista distribution, for your convenience in referring to the the source code and sample data files:

  setenv VISTA /u/cs449/src

DOCUMENTATION

The documentation for Vista is available in a considerable number of online manual pages. Start by looking at the following manual pages to get a quick overview:

Vista(7Vi) Introduction to Vista, including file formats and conventions.

Vindex(7Vi) A list of the currently available program modules and other manual pages.

These two manual pages should give you enough background to understand the examples given in the next section. Of course, you will also want to look at the manual page for each program module that you are using.

EXAMPLES OF USING VISTA

Following are some examples that you can run to demonstrate various existing Vista modules. See the manual page for each command for further details. Most Vista modules operate as ordinary Unix filters and take data from the standard input stream and place it on the standard output stream, while also allowing file names to be specified on the command line. See the Vista manual page for full details on command line conventions.

Displaying an image:

% vxviewe $VISTA/data/einstein.v

This program will pop up an X Window containing the first image from a Vista file. The program can be run on monochrome or color displays. The window can be resized in the same way as any other X window, and the image will be resampled to fit (maintaining the correct aspect ratio). Type Ctrl-N to see the next image in the file. See the manual page for information on how to zoom and display pixel values. Also, see the program **vxview** for a viewer with many more features.

Looking at the header of an image file:

```
% more $VISTA/data/einstein.v
```

The Vista file header is in ASCII and should be quite easy to understand. Full details are given in the **Vfile**(5Vi) and **VImage**(3Vi) manual pages. The binary data follows a new page character, so you can quit **more** before displaying the binary data.

Output an image to a printer using PostScript:

```
% vtops $VISTA/data/einstein.v | lpr
```

Run the **vtops** program to output an image to a printer in PostScript. This program will also output edge data files and can handle colour images and printers. It outputs encapsulated PostScript that can be included as a figure in a document.

Convolving an image with a convolution mask:

```
% vconvolve -mask mask.v <input.v >output.v
```

There is a sample mask for a 3x3 Laplacian operator in **$VISTA/data/laplace.v**.

Converting other image formats into Vista format:

```
% vconvert <image.iff >image.v
% pnmtov <image.pgm >image.v
```

Vista is upward compatible with the ‘‘IFF’’ image format used previously in the Laboratory for Computational Intelligence, so all programs will read images in the old format. For explicit conversion, use the **vconvert** program. The **pnmtov** program will convert images in the portable PBM, PGM, and PNM formats into the Vista format. Use **vtopgm** to convert in the other direction.

Detecting edges and linking them into edge lists:

```
% vcanny image.v | vlink >edges.v
```

vxviewe can be used to display edge sets overlayed on images from which they were obtained. This can be done by running **vxviewe** on both files and typing Ctrl-N to display the second file of edges on top of the image:

```
% vxviewe image.v edges.v
```

WRITING YOUR OWN PROGRAMS

Vista has been specifically designed to make it as simple as possible to add new modules to the system that integrate fully with all existing code. All source code for the system is available free to the public, so you can take the system to other sites and use whatever modules you have written. A particularly useful piece of source code is the **VImageView** widget that can display an image in a device- and size-independent way in an X window.

There is a 30-page introduction to Vista programming in the directory **$VISTA/doc/intro/**. The existing source code is in the **$VISTA/src** directory.

There are some sample modules available in the **$VISTA/examples** directory. You can copy one of these examples into your own directory, complete with **README** and **Makefile**, and modify it to create your own program. For example:

> **% cp -r $VISTA/examples/ramp** *my-directory*

There are extensive manual pages available to describe the various library routines and conventions used in Vista modules. You should read the **Vlib**(7Vi) man page first, as this will tell you about some issues, such as the C data types used in Vista programs, that you will need to know immediately.

Following is a list of some manual pages for the Vista programmer in rough order of importance:

Introduction:	**Vista**(7Vi), **Vlib**(7Vi)
File formats:	**Vfile**(5Vi), **VImage**(3Vi)
Programming conventions:	**Vattribute**(3Vi), **Voption**(3Vi)
Utility routines:	**VError**(3Vi), **VWarning**(3Vi), **VMalloc**(3Vi), **VNew**(3Vi), **VGetAttr**(3Vi), **VCreateAttrList**(3Vi)
Display widgets:	**VImageView**(3Vi)
Image handling routines:	**VCreateImage**(3Vi), **VCreateImageLike**(3Vi), **VCopyImage**(3Vi)
File I/O:	**VReadFile**(3Vi), **VWriteFile**(3Vi), **VReadImages**(3Vi), **VWriteImages**(3Vi)

THE VX LIBRARY

Programming interactive graphical programs for the X Window system has a very steep learning curve. Vista includes a library called VX that automates the task of creating X Window applications that display images and lines, incorporate menus and dialog boxes, and allow for mouse interaction. See the **VX**(3Vi) manual page for further details.

AUTHOR

David Lowe <lowe@cs.ubc.ca>