MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Introduction to the Diamond3D Vision Library

Paul Beardsley

TR2003-02 January 2003

Abstract

This report is an introduction to the Diamond3D computer vision library. The library enables fast creation of computer vision applications. Regarding the GUI and front-end of an application, there is support for image display, 3D display, mouse input to interact with images and 3D data, infrastructure for reading images from various sources, and for outputting images and derived data to disk. Regarding the main processing of an application, the architecture supports the creation of computer vision black-boxes which can be swapped between applications, allowing reuse with minimal effort. Furthermore, the libraryś specific goal is to enable creation of computer vision applicaitons with a focus on 3D geometry - the vision functionality includes feature detection, camera calibration, and multi-view geometry. The purpose of this report is to provide an introduction to the library. It is not a developersǵuide, but a developer should begin here.

Created March 2003.

Introduction to the Diamond3D Vision Library
Technical Report TR2003/02

Paul Beardsley

Mitsubishi Electric Research Laboratories,
201 Broadway, Cambridge MA 02139, USA
pab@merl.com

Abstract. This report is an introduction to the Diamond3D computer vision library. The library enables fast creation of computer vision applications.
Regarding the GUI and front-end of an application, there is support for image display, 3D display, mouse input to interact with images and 3D data, infrastructure for reading images from various sources, and for outputting images and derived data to disk. Regarding the main processing of an application, the architecture supports the creation of computer vision black-boxes which can be swapped between applications, allowing reuse with minimal effort. Furthermore, the library's specific goal is to enable creation of computer vision applications with a focus on 3D geometry - the vision functionality includes feature detection, camera calibration, and multi-view geometry.
The purpose of this report is to provide an introduction to the library. It is not a developers' guide, but a developer should begin here.

1 Introduction

This report is an introduction to the Diamond3D library. A distribution of the software has two parts - the **Diamond3D** directory contains applications and the library; the **tools** directory contains the Makefiles for building applications. Assuming the user already has these directories set up, this report describes how to run an application and the most common GUI interactions.

2 Running an Application

Applications run under Linux and Windows. The directory for an application <myapp> is Diamond3D/App/<myapp>. To run an application under Linux, or in a cygwin shell under Windows, first set the environment variable MERL_TOOLS to the tools directory. Then do

cd Diamond3D/App/<myapp>/Fltk

To run a debug version of the application, do

make run

To run a release version, do

make Config=gcc2-release run

An application which reads images from disk, or which is saving images to disk, will have an associated 'dataset' directory. See Section 5 about datasets. The user sets up a dataset directory explicitly by doing

mkdir <someplace>/<my-dataset>
mkdir <someplace>/<my-dataset>/Image

There are other sub-directories in the dataset which are required when the user is creating and storing derived data from the images. The additional directories are listed in Section 5

The user can select a dataset in two ways. Firstly, at run-time, by invoking a directory browser (see Section 3). Secondly, a default dataset can be specified by creating a file Diamond3D/App/<myapp>/AppData/appdata.dirname with the text

<someplace>/<my-dataset>/

A default dataset is more convenient than using the directory browser if one is running an application with the same dataset over and over again.

3 User Interface

Application GUIs are created using the Fltk toolkit (www.fltk.org)[1]. Most of the terminology for widgets in the remainder of this section is obvious, but the Fltk term 'checkbox' is used for a toggle button with visual indicator of the current state on/off.

The 'Grab' application is an image viewer which illustrates many of the functions which are common across all Diamond3D applications. Figure 1 shows the Grab GUI. The window at left is the full-size image display. The window at right is a zoom-display.

The remainder of the section lists Grab's individual components and mouse functionality. Note that sometimes checkboxes and buttons can have the same label (e.g. 'Output memory'), but the subsection headers below serve to differentiate them.

3.1 Menu - Session / Choose directory

This pops up a directory browser to allow selection of a dataset. The application is placed into movie mode and loops through sequential read-and-display of images in the dataset.

[1] This toolkit is free, is cross-platform, and the data files for the GUI are created in ASCII so that a user can edit them directly which is occasionally convenient.

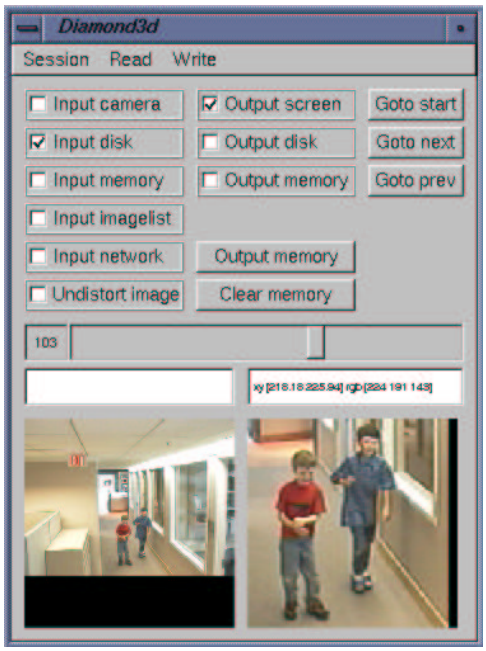

Fig. 1. Example Diamond3D application - the Grab image viewer.

3.2 Menu - Session / Exit

This terminates the application.

3.3 Menu - Read / Read image memory

This reads the image sequence for the current dataset into memory. The application is placed into movie mode and loops through sequential display for the images in memory. The dataset directory must be selected prior to this by using [Session/Choose directory], or by using a default dataset.

3.4 Menu - Write / Write image memory

This writes the image sequence which is currently in memory to disk. The dataset directory must be selected prior to this by using [Session/Choose directory], or by using a default dataset.

3.5 Checkbox - Input camera

This toggles input from the camera. If the application has not been compiled with a camera, or a camera is attached but is not returning data, an error is output and no action is taken. Otherwise the application is placed into movie mode and loops through sequential grab-and-display for camera images.

3.6 Checkbox - Input disk

This toggles input from disk for images in the current dataset. The dataset must be selected prior to this by using [Session/Choose directory], or by using a default dataset.

3.7 Checkbox - Input memory

This toggles input from memory for images which are currently in memory. Images can have been placed in memory prior to this by [Read/Read image memory], or checkbox [Output memory], or button [Output memory].

3.8 Checkbox - Input imagelist

This toggles input from an explicit list of named image files. The checkbox pops up a file browser to allow selection of a file whose contents have the form

image-file1
image-file2
...

The application is placed into movie mode and loops through sequential display for the named images.

3.9 Checkbox - Input network

For MERL internal distributions, this checkbox toggles input from the CHOnet camera network. **Implementation note:** not available at March 2003.

3.10 Checkbox - Undistort image

This toggles undistortion mode for the images. If undistortion is turned on, the application reads a lens calibration file for the current dataset from disk, and use this information to resample input images to an undistorted form. **Implementation note:** not available at March 2003.

3.11 Checkbox - Output screen

This toggles all output to the screen. It can be used to disable output when doing timing tests on application code.

3.12 Checkbox - Output disk

This toggles output of images to disk. A dataset must be selected prior to this by using [Session/Choose directory], or by using a default dataset. Typical usage is to activate [Input camera], and activate [Output disk] to store a live sequence to disk.

3.13 Checkbox - Output memory

This toggles output of images to image memory. Typical usage is as [Output disk].

3.14 Button - Output memory

Store the current image into image memory.

3.15 Button - Clear memory

Clears the image memory.

3.16 Button - Goto start / Goto next / Goto prev

Go to the first/next/previous image in the current dataset.

3.17 Slider - image sequence

The image sequence slider allows navigation of fixed image sequences, for example seqquences being input by [Input disk] or [Input memory]. It is inoperative with [Input camera]. The left-side of the slider shows the index of the current image, $0-(n-1)$, where n is the number of images in the sequence. Apart from using the slider to actively navigate the sequence, its position reflects changes due to other events such as [Goto next].

3.18 Textfield - image source

This textfield displays information about the current image source - e.g. the name of the camera, or the name of the dataset directory. **Implementation note:** not available at March 2003.

3.19 Textfield - image pixels

This textfield displays information about the current pixel - use centre mouse button hold-and-drag to specify a pixel in the image.

3.20 Image Display - Mouse Operations

All mouse operations work on the main display and on the zoom display. Points and line segments can be created on the image plane to provide input data for any applications which require them.

- Left-button down-and-release to specify an input-point.
- Left-button hold-and-drag to specify a zoom area.
- Centre-button hold-and-drag to see pixel values.
- Right-button hold-and-drag to specify a line segment.
- (Centre-button will in future allow object-move and vertex-move operations for overlay items on the image).

4 Further User Interface

This section lists GUI functionality which is not in 'Grab' but which is common to many other Diamond3D applications.

4.1 Checkbox - Output graphics

It is common for Diamond3D applications to overlay images with derived data e.g. an edge map. These images-plus-overlay can be stored to disk, and a full sequence of the stored images then made into a movie to show the operation of an application.

Typical usage is to activate an image input, and to activate some vision processing, and activate [Output graphics] to write the results to disk. This will write images for all the display areas on the GUI - main displays and zoom displays, and the 3D display if there is one.

4.2 3D Display - Mouse Operations

- Left-button to rotate the 3D data.
- Centre-button to translate the 3D data.
- Right-button in a radial direction to zoom.

5 Datasets

A dataset directory contains the following sub-directories.

- **Image** contains images. The standard filestem for an image sequence is 'seq'. The images are stored separately as seq.000.PNG, seq.001.PNG etc. The name can be customised using the file Param/image.filestem e.g. if this file contains the string 'myimages', then the application assumes images with name myimages.000.PNG, myimages.001.PNG etc. Stereo sets are stored as seq.000.s0.PNG, seq.000.s1.PNG etc. Images are read using the ImageMagick filter which can handle most image types - contact MERL if a required image type is not handled.
- **Data*** are directories containing data created by the application.
  - **DataAscii** contains ASCII data files.
  - **DataLinux/DataWindows/DataIrix** contain binary data files. The appropriate directory is accessed automatically by the application according to the current platform.
- **Param** contains ASCII files for application parameters, created and edited by the user.
- **ImageData** contains any derived images generated by the application. See Section 3, [Output graphics].

6 Input from Mono and Stereo Image Sources

A mono application like Grab can read from a stereo image source. If the user chooses a stereo image source, then a command-line prompt asks the user to specify one of the stereo cameras for input. In fact, any Diamond3D application with an m-camera GUI can read images from an n-camera image source, where $m <= n$. Any subsequent file operations like [Output disk] will also associate the images with the specific stereo camera e.g. [Input camera] for a stereo camera in conjunction with [Output disk] will write image files to disk with appropriate naming for the current stereo camera.

7 Developer Documentation

See [1] for developer information. See [2] for information about attaching a camera to a Diamond3D application. Source code documentation is in Diamond3D/Doc. For application <myapp>, the documentation is

- Diamond3D/Doc/App-<myapp>/html/index.html
- Diamond3D/Doc/App-<myapp>/latex

For library d3d-<somelib>, the documentation is

- Diamond3D/Doc/d3d-<somelib>/html/index.html
- Diamond3D/Doc/d3d-<somelib>/latex

References

1. P.A. Beardsley. Developer guide to the Diamond3D Vision Library. In preparation March03, MERL, 2003.
2. P.A. Beardsley. Using cameras with Diamond3D applications. In preparation March03, MERL, 2003.