A prototype of a beam steering assistant tool for accelerator operations

M. Bickley and P. Chevtsov, Jefferson Lab, Newport News, VA 23606, USA

Abstract

The CEBAF accelerator provides nuclear physics experiments at Jefferson Lab with high quality electron beams. Three experimental end stations can simultaneously receive the beams with different energies and intensities. For each operational mode, the accelerator setup procedures are complicated and require very careful checking of beam spot sizes and positions on multiple beam viewers. To simplify these procedures and make them reproducible, a beam steering assistant GUI tool has been created. The tool is implemented as a multi-window control screen. The screen has an interactive graphical object window, which is an overlay on top of a digitized live video image from a beam viewer. It allows a user to easily create and edit any graphical objects consisting of text, ellipses, and lines, right above the live beam viewer image and then save them in a file that is called a beam steering template. The template can show, for example, the area within which the beam must always be on the viewer. Later, this template can be loaded in the interactive graphical object window to help accelerator operators steer the beam to the specified area on the viewer.

INTRODUCTION

The CEBAF accelerator provides nuclear physics experiments at Jefferson Lab with high quality electron beams. High quality of the beam means its precise position on a target, almost ideal gaussian profile, and very small ($\sim 2 \cdot 10^{-5}$) relative energy spread. All of this is extremely important for the nuclear physics program at Jefferson Lab to advance human understanding of the atom's nucleus.

Visual beam quality control at Jefferson Lab is done with the use of two different types of beam viewers: "classic" and "direct light" beam viewers.

"Classic" beam viewers require thin luminescence screens (for conventional viewers) or carbon foils (for optical transition radiation monitors or OTR) to be inserted in the beam. As a result, the "classic" viewers are invasive and can only be used at relatively small beam currents.

"Direct light" beam viewers at Jefferson Lab are synchrotron light monitors (SLM) and synchrotron light interferometers (SLI). They are based on the use of synchrotron light generated by relativistic electrons moving in the magnetic fields of dipole magnets. The "direct light" beam viewers are absolutely not destructive (not invasive) for the beam.

The resultant beam images from the viewers are captured by TV cameras and displayed in the accelerator control room on the main control display wall as well as on numerous TV screens all over Jefferson Lab. The TV signals are also fed into pipelined high performance image processing systems Maxvideo 200 [1]. The main advantage of the pipeline technology is that the pixel manipulation can be done while the image is being digitized and directed to the image memory. As a result basic image processing operations can be implemented at the full 30 Hz frame rate of the standard NTSC video signal.

MAXVIDEO APPLICATIONS IN ACCELERATOR CONTROLS

All beam image analysis applications at Jefferson Lab are based on the information provided by Maxvideo systems. The systems run a large amount of control and image processing software that has been continuously updated to meet the needs of accelerator operations. The software routinely performs such important functions as masking the pixels outside of the region of interest, subtracting a background image, estimating the transverse RMS beam size and many others. Beam characteristics together with the digitized beam images are entered into a control system database that makes them available for any application running on the accelerator control computer network.

A large variety of high level applications (these applications usually run on workstations) and scripts created at Jefferson Lab, based on Maxvideo systems, allow the users, for example,

- to have a live beam image from any viewer on a monitor of any computer or X terminal connected to the accelerator control network (we call this X-windows application a "beam movie");
- to make a snapshot of any beam image, publish it in an electronic logbook [2], and save it in a file for future reference;
- to calculate the beam energy spread on the basis of the SLI interference pattern and data model as well as to estimate the calculation errors and the data model reliability [3].

Beam image analysis applications significantly simplify beam tuning and diagnostics tasks for the accelerator operations specialists and contribute to very high beam availability for nuclear physics experiments at Jefferson Lab.

Notice: Authored by The Southeastern Universities Research Association, Inc. under U.S. DOE Contract No. DE-AC05-84150. The U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce this manuscript for U.S. Government purposes.

At the same time, adding to these applications some functions, which are not typical for “classic” accelerator control software, can significantly increase this contribution.

For example, consider CEBAF accelerator setup procedures. For each machine operational mode, these procedures are very complicated and require extremely careful checking of beam spot sizes and positions on multiple beam viewers. It would be very helpful to have a graphical application that allows the accelerator operations crew to draw simple geometric objects right on top of a live viewer beam image. Such a drawing can show, for instance, the area within which the beam must always be (or not be) on the viewer and become a beam steering template. Once created and saved in a file, such a template can later be used as a reference picture for steering the beam to (or out of) the specified area on the viewer.

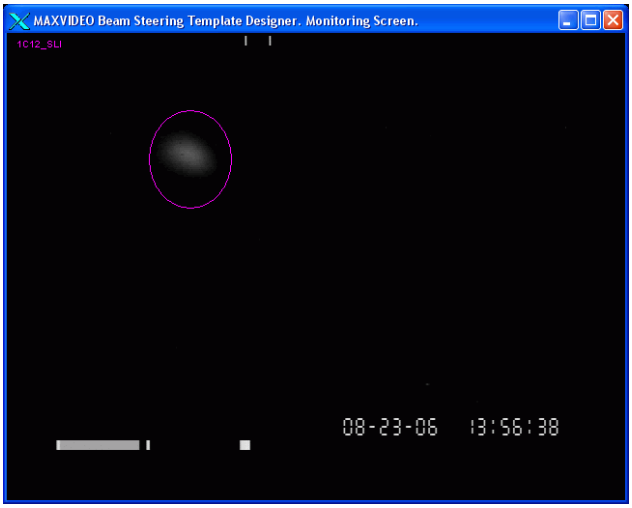

Figure 1: Monitoring screen of the Beam Steering Template Designer application.

If we look for commercial and non-commercial painting software packages available in the market then we find that some of them allow for drawing graphical objects directly over the frames in video sequences. The effects of the use of these packages are amazing. For example, the CinePaint tool [4] was used on The Last Samurai feature film to add hundreds of flying arrows to a battle scene. The main problem is to integrate any of these packages in the control system environment. It is evident that this problem is much more complicated than creating a relatively simple drawing tool as a part of the existing accelerator control and video image processing systems. A prototype of such a tool was created at Jefferson Lab in a few weeks. We call it a Beam Steering Assistant Tool.

The Beam Steering Assistant Tool has been developed on the basis of the Qt software package [5]. It is computer platform independent, multi-threaded, and consists of two graphical user interface (GUI) applications: the Beam Steering Assistant and the Beam Steering Template Designer.

BEAM STEERING TEMPLATE DESIGNER APPLICATION

The main purpose of the Beam Steering Template Designer application is to create and save beam steering templates for each beam viewer and each machine operational mode. The application provides a user with two information windows: a monitoring screen and a control panel.

The monitoring screen (Fig. 1) has an interactive graphical object window, which is an overlay on top of a digitized live beam viewer image. With the use of a computer mouse, one can easily create and destroy simple color graphical objects in this window. The default object color is magenta. It can be changed when the application starts. The upper left corner of the monitoring screen shows the name of the beam viewer that is currently being used in accelerator operations.

Object manipulations in the interactive graphical object window are handled by the control panel (Fig.2). The panel has information about all functions implemented in the Beam Steering Template Designer application.

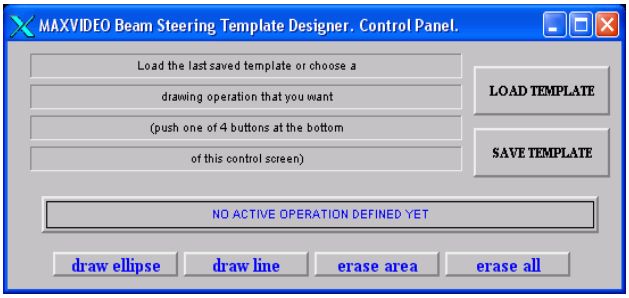

Figure 2: Control panel of the Beam Steering Template Designer application at its startup.

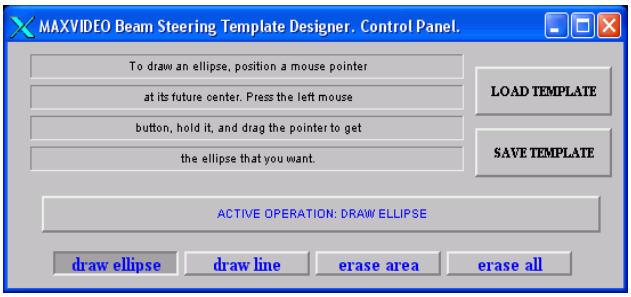

Figure 3: Control panel at work.

When the Beam Steering Template Designer application starts, its control panel looks exactly like one represented in Fig. 2. The bottom part of the panel shows that at this point one can draw ellipses and lines as well as erase created graphics in any area of the window or in the entire window at once.

One can also save graphical objects in a file and load previously created graphics or a beam steering template from a previously saved file. The central status line shows the operation that is currently active. The upper left part of the panel informs the user about what can be done for each active operation or how to activate graphics.

The Beam Steering Template Designer is very easy to use. For example, to draw an ellipse above a live image, you have to press the “draw ellipse” button on the control panel. The panel immediately responds to this command providing the information about what you are going to do (“ACTIVE OPERATION: DRAW ELLIPSE” in this case) and how to do it (see Fig. 3 for details).

We note that, for example, each ellipse is drawn by positioning a mouse pointer at its future center and then dragging the pointer to get the required shape and size. This allows for easy creation of graphical objects with the reference to the beam spot locations on a viewer as it can be seen in Fig. 1.

The software that handles the information windows of the Beam Steering Template Designer is written in C++. It consists of two threads running simultaneously: a main thread and an information thread.

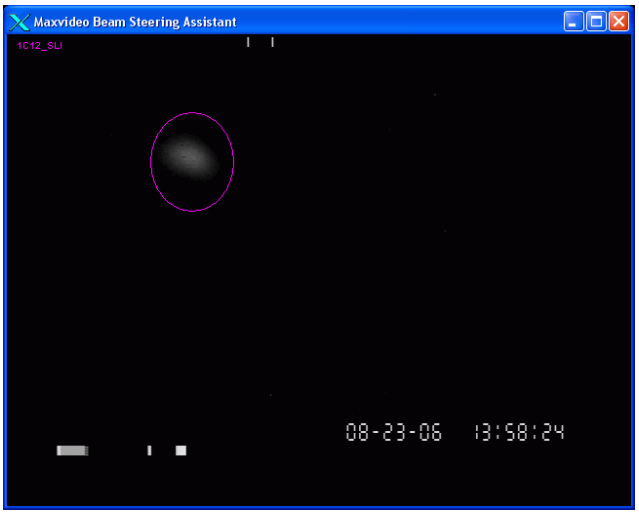

Figure 4: Main information screen of the Beam Steering Assistant application.

The information thread continuously updates the beam viewer images (provided by the Maxvideo systems and control system database) in the graphical object window. It also takes care of the name of the currently used (active) beam viewer, which is shown in this window. Information about active viewers and beam viewer images is a part of the control system database at Jefferson Lab.

The main thread of the software implements the user interface. It handles all the functions of creating and destroying graphics in the graphical object window as well as saving created graphical objects as a beam steering template in a file and loading previously created templates in the window.

Once created and saved, each template can later be used to reproduce the accelerator mode for which this template was created by steering the beam to the specified area on the viewer. We note that for this job the graphics functions of the Beam Steering Designer are not used at all. That is why a light version of this application was created. We call it a Beam Steering Assistant application.

BEAM STEERING ASSISTANT APPLICATION

The Beam Steering Assistant application has only one information window, shown in Fig. 4. During the startup the application requests the information about the beam steering template for the active viewer to use (the default is the last created) and the color to draw this template (the default is magenta) on top of a video image.

The software handling the information window is written in C++ and consists of two threads: a main or viewer image thread and a viewer name thread.

The main thread continuously refreshes beam viewer images in the information window on the basis of the Maxvideo data.

The viewer name thread takes care of the name of the active accelerator beam viewer, which is shown in the upper left corner of the information window. When the active viewer changes, the application updates this name and loads the last created template for this viewer.

In the latest version of software, before loading a template the application pops up an additional small window containing the list of all templates available for this viewer and asks users to choose one. This makes it much easier for users to switch to any desired accelerator mode on the fly.

CONCLUSION

A prototype of a Beam Steering Assistant tool created at Jefferson Lab is an example of implementing some “non-typical” functions in control and data processing systems. The tool, which looks like a simple computer entertaining program allowing users to draw various graphical objects over live TV images, can have a very positive impact on accelerator operations. In particular, it can help simplify accelerator setup procedures and make them reproducible.

REFERENCES

[1] P. Chevtsov, “Multivideo Source System for Beam Diagnostic Applications” PCaPAC-2000. DESY. Hamburg. Germany. 2000.

[2] T.Larrieu, T. Mcguckin, “Beyond an Electronic Logbook”. PCaPAC-2005. Hayama. Japan. 2005.

[3] P. Chevtsov, “Automated Image Quality Optimization for Synchrotron Light Interferometers”, ICALEPCS 2005, Geneva, Switzerland, 2005.

[4] www.cinepaint.org

[5] www.trolltech.com