Integration of Camera Systems and Investigation of Optic Flow

Evan Kahn
Dr. Adrienne J. Raglin and Dr. David A. Ligon
Army Research Lab

Abstract

The original aims of this study were to allow a Defend-IR DI-5000 infrared/visible/laser rangefinder assembly to interface with a computer through a portable, user-friendly Java application with flexible input methods, and to allow the user to capture from, as well as control, the camera. Since the DI-5000 assembly at hand lacked the integrated Joystick Control Unit, an alternate piece of software was commissioned, which allowed a user to control the DI-5000 with the hexadecimal codes sent via serial port commands documented in the camera's specifications. Longer-term goals involve the creation of software in MATLAB to calculate and adjust for the motion of the camera. This allows moving objects in the frame to be analyzed from a simulated, motionless viewpoint. With an artificially motionless point of view, we can then calculate the image flow of objects moving in the frame. Using the DI-5000's built-in laser rangefinder, we can also calculate the distance from those objects as well as their speed.

Introduction / Background

The long-term aims of this study included facilitating the detection and tracking of moving objects through the use of automated tracking software that receives information from multiple articulated cameras equipped with laser rangefinders. Eventually, optic flow principles would be used to calculate the location of, and distance to, moving objects. This is useful for future automated systems designed to detect and track down moving objects. The technology required to allow a camera system to intelligently follow a moving object as well as compensate for its own motion is rooted in the principles of optic flow, first described in the 1940s. Optic flow is a field of study that describes how motion is perceived in the two-dimensional 'image plane': to a single human eye as well as to a camera.

However, time constraints meant that the complete goal of this project was never achieved. Instead, optical flow processing software developed by Visesh Chari and uploaded to MATLAB MathWorks was used to analyze the images output from the software.

Materials and Methods

A Defend-IR DI-5000 camera, sourced from the Army Research Lab, was used for the object tracking. Java software enabled the user to perform a number of vital functions, including panning, focusing, and tilting the camera, switching between the camera's visible and infrared sensors, and operating the camera's laser rangefinder. The camera is connected to the computer via a VPort Ethernet interface, and the camera is controlled through the computer's serial port.

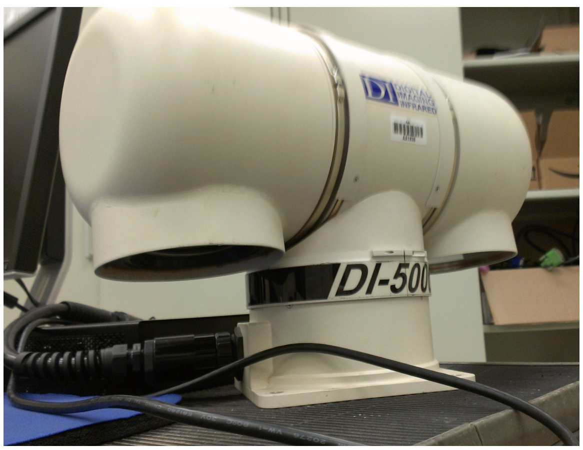

Fig. 1: The Defend-IR DI-5000 camera.

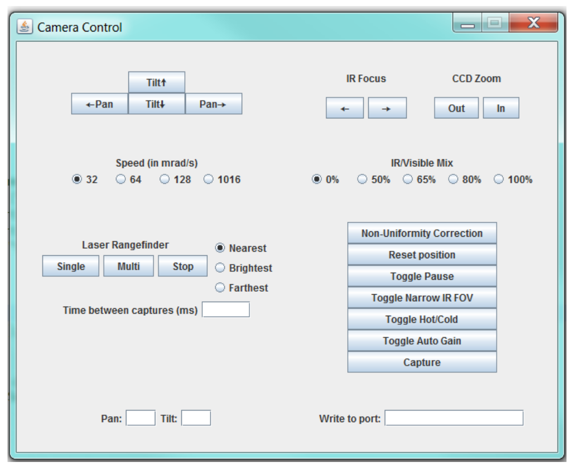

Fig. 2: The software created to control the camera.

Serial Communication

The bulk of this project consisted of writing Java-based GUI software to control and capture images from the DI-5000 camera. Various hexadecimal codes were pulled from the documentation that came with the camera, and the RXTX extension for Java was implemented to allow the camera to communicate with the computer.

The camera involved had a number of mechanical adjustments that needed to be controlled via software. Typically a standalone joystick unit would be connected to the camera via its serial connection in order to control its rotation, tilt, zoom, and focus. In the absence of such a joystick, a program was created to interface with the camera.

In order for the camera to understand the commands sent by the user, the program needed to generate specific hexadecimal values to be sent via the serial port. For example, in order to rotate the camera 30 degrees clockwise, the number 30, after being input by the user, was first converted from degrees to milliradians and then converted to a hexadecimal value. Additional steps were taken to ensure that the number input was appropriate for conversion - for instance, the camera's serial interface accepts different formats depending on whether the input value is positive or negative. Additional controls (corresponding to hexadecimal values hardcoded in the serial interface) included various image adjustments such as non-uniformity correction (the camera takes a series of images to automatically calibrate its brightness settings). Also built into the software was a panel to allow the operation of the laser rangefinder – which will prove useful in future applications (see Conclusions section).

Image Capture

Initially, images were captured through a digital video recording device that made use of the camera's analog video out port. However, this only supplied a constant video stream and provided no way to capture individual images. As an alternative the VPort 2310 Ethernet device was used. This device allowed still image capture over a LAN between the Ethernet-connected VPort and the computer. The Java Authenticator class was used to allow the user to access the VPort software over the network.

Image capture, which was consistently successful following the installation of the VPort module, allowed images to be taken on command or over a brief time interval. The initial intent was to create an original piece of optic flow software, but a number of issues stemming from the low resolution of the images and the difficulty of automating the comparison between images, led us to use an established optic flow calculation solution.

Software by Visesh Chari of the International Institute of Information Technology, Hyderabad, sourced from the Matlab MathWorks database, was used to facilitate optic flow calculations. The software package, entitled "High Accuracy Optic Flow", included two software algorithms designed to calculate flow: the Brox algorithm, originally detailed by Thomas Brox in 2004, and the Sand algorithm, originally detailed by Peter Sand. We chose to use the Brox algorithm, which provides good reliability and is highly configurable.

A number of images were captured with the camera array, indoors and outdoors, focusing on both stationary objects and environmental features, and moving objects. The images on which we began analysis were taken from a set including a moving truck and the stationary background against which that truck was moving. In order to establish some degree of mobility, the camera, the PC used to control it, and the PC's peripherals were set up on a rolling cart and plugged into a UPS (uninterrupted power supply, with an integrated battery) in order to permit untethered operation. The two CRT monitors depicted allowed the user to view a live analog preview directly from the camera output, unhindered by the VPort unit's digital encoding process. In order to maximize longevity, the computer used was a low-powered PC with solid-state memory, running Windows XP.

Fig. 3: The camera setup used in the field.

Results and Conclusions

Simply put, optic flow principles serve to determine the continuity of features in a series of images that represent motion: the edges and corners detected in one image cross over to the next, albeit in different locations. The assumption that the overall "flow" of the image remains constant – that is, that everything moving from one image to the next does so in a uniform and predictable manner – can be expressed in a graph of all optic flow that occurs in the image. Since the graph in question comes from the two images shown – including a moving truck and a moving camera – the outline of the truck is seen in the flow as well as the more uniform arrows indicative of the moving point of reference.

Fig. 4: Four frames of an image capture including both a moving object and a moving point of reference.

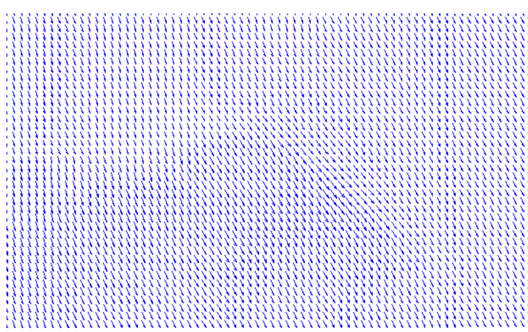

Fig. 5: An enlarged portion of the optic flow diagram yielded via the Brox algorithm, applied to the first two 'truck' images shown above.

The final camera control software consisted of a GUI-based control system including four directional controls allowing the user to pan the camera clockwise and counterclockwise, as well as tilt up and down. It was also possible for the user to set the camera to pan and tilt to a specific degree measurement on each axis. The laser rangefinder could be triggered and set to take a different number of measurements at a time, and data can be sent and received simultaneously through the serial port (although the DI-5000 did not support this feature). Images could be captured individually or at an interval of a user-specified number of milliseconds. The infrared lens could be manually focused, and the visible lens can be zoomed in and out. If the user required a superimposition of the visible and IR camera output, it was possible to set the 'visible-IR mix' to any point between 100% visible and 100% infrared, in increments of 25%.

While the creation of the camera control software (detailed in more depth in the Materials and Methods section) was successful, due to time constraints the fullest exploration of the optic flow processing software proved inconclusive. Its ability to trace major features between images seemed to work properly, but its extreme lack of speed (comparing two frames often took over 20 minutes) made it difficult to evaluate. Further research warrants exploration of the practical applications of these algorithms, including the tracking of moving objects and the automated slaving of a group of cameras to said object using basic geometry in tandem with the optic flow algorithms. For instance, using the laser rangefinder present on the DI-5000, combined with optic flow principles and geometry, could enable the user to determine the exact size, as well as speed, of a distant object. This could prove invaluable for defense applications as well as in civilian and consumer fields.

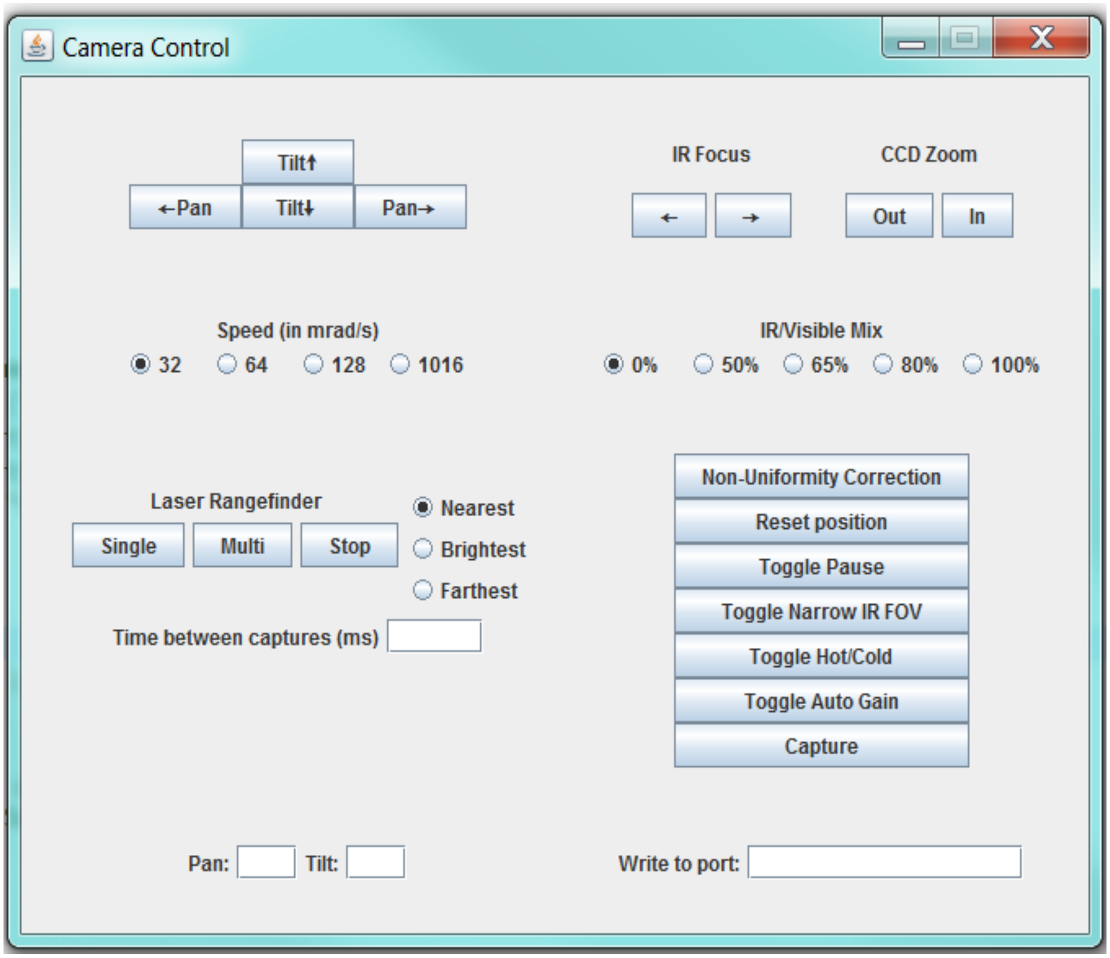

Fig. 6: The final version of the software used to control the camera apparatus.

References

DefendIR integrated thermal/ccd/ptz camera. (n.d.).

Retrieved from http://gs.flir.com/uploads/file/products/brochures/defendir.pdf

Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). *High accuracy optic flow estimation based on a theory for warping*. Manuscript submitted for publication, Mathematical Image Analysis Department, Saarland University, Saarbruücken, Germany. Retrieved from http://lmb.informatik.uni-freiburg.de/Publications/2004/Bro04a/brox_eccv04_of.pdf

Owens, R. (1997, Oct 29). *Optical flow*. Retrieved from http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT12/node4.html

Chari, V. (2007). *Optical flow computation*. Informally published manuscript, Retrieved from http://perception.inrialpes.fr/~chari/myweb/Software/flow_documentation.pdf