

# T-1001

## AUTONOMOUS BATTLE ROBOT SPECIAL SENSOR REPORT

FERNANDO HERNANDEZ  
8/01/2005

— TAs —

WILLIAM DUBEL  
STEVEN PICKLES

— INSTRUCTORS —

A. A. ARROYO  
E. M. SCHWARTZ

— UNIVERSITY OF FLORIDA —

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
EEL 5666  
INTELLIGENT MACHINES DESIGN LABORATORY

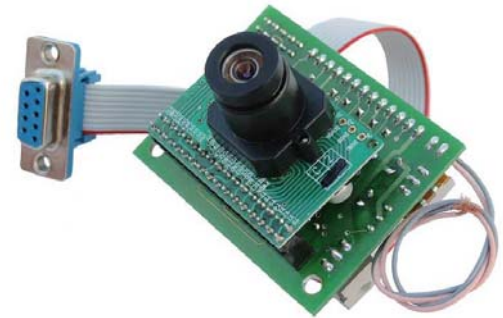
*Abstract:*

*The T-1001 is an autonomous “battle tank” robot that patrols a battle zone, fires on enemy units, and avoids friendly units. This tank distinguishes enemy from friend by color, firing on enemy units when encountered. In addition to simply firing, the T-1001 keeps track of the amount of ammunition that it has left, and when the ammo has run out the tank will switch into “battering ram” mode, in which it will ram the enemy instead of firing upon it. In addition to these two behaviors, the T-1001 will detect a black line on the floor marking the limit of the battle zone, and not wander beyond it (No retreat! No surrender!). The T-1001 uses sensors of the following four types: ultrasonic (Devantech SRF04), IR (Lynxmotion IRPD-01), photo reflector (Hamamatsu P5587), and vision (CMUCam 1) to perform its functions and to properly interpret the battle zone.*

# **CMUCAM SPECIAL SENSOR**

---

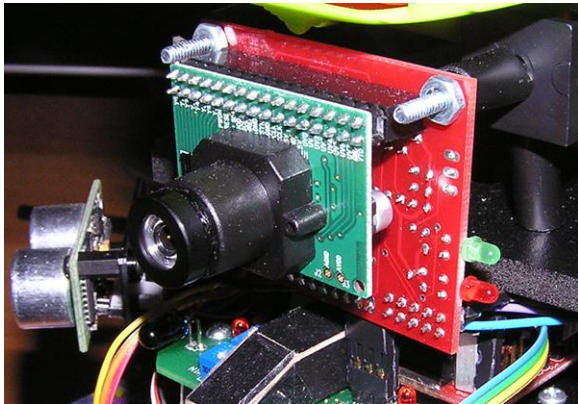
In order to detect the enemy and properly navigate on the battle field, the T-1001 implements various sensors, of which the CMUCam is the most complex. This report details the workings and my experience with the CMUCam during the course of building and programming the T-1001 autonomous battle robot.



## **ABOUT THE CMUCAM**

The CMUCam vision sensor is made by a group of people at Carnegie Mellon University, hence its name. The CMUCam is a great vision sensor for applications where simple low level sensors like CDS cells do not provide enough data about the environment. This camera performs much of its processing on board, communicating statistics of the images that it sees to the microcontroller board attached to it at up to 17 frames per second. This allows a small microcontroller (which would normally not be capable of dealing with vision) to use vision to interpret its environment. This means that now the realm of vision sensing is no longer out of reach of the casual robotics hobbyist.

## HOW IT WAS USED



The T-1001 uses the CMUCam to orient itself on the battlefield and to detect enemies so that it can fire upon them. The camera is mounted facing forward, in order to detect enemy targets that cross the robot's path. As the robot patrols, the camera continuously scans for red objects (which are considered enemies). It is given a slight downwards tilting, to prevent the robot from seeing too far into the distance and possibly picking up interference from objects in the background. An added measure to prevent the robot from picking up any erroneous target information from red objects in the background was to limit the visible area to the lower 45% of the screen, making the effective enemy detection range of the robot roughly 42 to 48 inches.

The camera has a large serial command set, the entirety of which can be found in the CMUCam documentation. Of interest are commands such as "GM" which returns the average value of red, green, and blue on the image, as well as their standard deviations, the TC command which is used to track a color and return its bounding box and its center coordinates, and the SW command which is used to set the current view window, allowing one to track colors only in a certain portion of the screen.

The CMUCam is interfaced to the Maverick board using UART1, which is the second RS-232 port on the board, and I have selected 38.4k baud as the baud rate. Since the UART ports on the board are not the standard pin headers but instead the screw-down clamp type, I have them go first to a corner of the power board, to where they connect to a pin header, into which the camera's pins are attached. This makes it easy to remove the camera, without needing a screwdriver to release the cables from the UART port.

## **SOFTWARE ALGORITHM**

In order to communicate with the CMUCam, I developed several simple functions with some assistance from Julio Suarez. The algorithm I used to control the robot using the CMUCam data is shown below (actual code found in code appendix of the main report):

- 1) Initialize the camera, set for poll mode, raw output, middle mass mode, and set the visible screen area to the bottom 45%.
- 2) Use the “Track Color” command to attempt to find a certain color in the camera’s field of vision.
- 3) If the color is found, call the track color command three more times and average out the readings for the center of the blob, in order to get a good idea as to where the object is located, then move the robot accordingly to center the object in the field of view, and move towards the object. If we are close enough to shot the target, we fire the main gun or ram it depending on how much ammo we have left.
- 4) If the color is not found, check the other sensors and respond accordingly.
- 5) Go to step 2.

As it can be seen from above, the algorithm is not complex at all, but I am confident in reporting that it works amazingly well.

## **MY THOUGHTS ON THE CMUCAM**

While the T-1001 autonomous battle robot would not have been possible without the CMUCam. The many features of the CMUCam make it deal for robotics projects similar to mine. While the camera is full of useful and needed features, it is important to note that the CMUCam itself has a little room for improvement. Below I list a few pros and cons based on my experience with the CMUCam vision sensor:

### **Pros -**

- Returns the center of a color blob when in middle mass mode
- Return specific RGB color values of any pixel
- Returns average RGB color values of any region of pixels
- Visible window can be dynamically set
- Auto white balance and auto gain can be turned on or off dynamically

### **Cons –**

- Comes bundled with very bad documentation (the online manual which I found after a few days of trying to decipher the paper manual was much more helpful)
- Pixel data tends to have a lot of red tint to it, causing large R values to be present in nearly every packet
- Returns an image which is off center to the right by about 10° (I had to compensate for this in software)
- Has an obtrusive voltage regulator sticking out the back of it, which makes mounting difficult (I bent it sideways and out, but now it sticks off to one side and gives the camera an asymmetric look to it)

All in all, I would say that the pros of the CMUCam definitely outweigh the cons, and that this is the camera of choice when affordable vision sensing is required. I would also like to add that while there exists a CMUCam 2, the superior frame rate and a few other new features that it provides may not even be used in many robotics projects. The features that the regular CMUCam provides are in many scenarios all that is needed in order to process a robot's visual environment.

## **REFERENCES / SOURCES**

---

NOTE: The following papers/websites were used during the making of my robot and the creation of this paper. I kindly acknowledge their assistance.

CMUCam User's Manual v2.00

Anthony Rowe and Carnegie Mellon University, edited by Charles Rosenberg and Illah Nourbakhsh

< <http://www-2.cs.cmu.edu/~cmucam/Downloads/CMUcamManual.pdf> >

MAVRIC-IIB Manual

< <http://www.bdmicro.com/images/mavric-iib.pdf> >

Atmega128 Complete Datasheet

< [http://www.atmel.com/dyn/resources/prod\\_documents/doc2467.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf) >

I would like to give a special thanks to Julio Suarez for assistance with the initialization and communication code for the CMUCam.