

Rescue Robot

Sunantha Krishnan

Department of Information Technology

Don Bosco Institute of Technology, Mumbai 400070, Maharashtra, India

Anthony Selva Jessobalan

Department of Information Technology

Don Bosco Institute of Technology, Mumbai 400070, Maharashtra, India

Abstract - We propose a Rescue Robot system which is a cross platform connection between robot and a PC. The Robot is designed with thermostat to sense temperature, camera to detect people in the rubbles and wheels for the movement, with end effector which can lift a weight of around 200 to 300 gm. Our robot will move in the debris to detect human by using Viola- Jones face detection algorithm. Totally the system aims in directing people among debris, which usually takes a day or so to bring the regular resources into picture; but our system can get the situation within hours which can aid in giving priority to people alive. It is fast, cost efficient and easy to use.

Keywords - rescue robot, cross platform, face detection

I. INTRODUCTION

Our system can be defined as a robot that has been designed for the purpose of helping rescue workers. Common situations that employ rescue robots are mining accidents, urban disasters, hostage situations, and explosions. Search and rescue technology to-date still rely on old technologies such as search dogs, and technology that has been in service for decades. Intelligent robots equipped with advanced sensors are attracting more and more attentions from researchers and rescuers. When a living thing tries for a rescue operation, the life of the rescuer will be highly at risk. In such situations, technology will be of utmost importance. Hence, we present our project, Rescue Robot to help such rescue operators. Our Rescue robot will serve as a trapped disaster victim's lifeline to the outside world.

II. SCOPE OF THE PROJECT

The use of heavy machinery is prohibited because they would destabilize the structure, risking the lives of rescuers and victims buried in the rubble. Only by hand should the pulverized concrete, glass, furniture and other debris be removed. Rescue specialists use trained search dogs, cameras and listening devices to search for victims from above ground. Though search dogs are effective in finding human underground, they are unable to provide a general description of the physical environment the victim locates. Camera mounted probes can provide search specialists a visual image beyond voids that dogs can navigate through, however their effective range is no more than 4-6 meters along a straight line below ground surface. The fire sensors have a range of 2 m. the RF pro has a range of 30 m along a straight line. Arduino UNO has a clock speed of 16 clocks. The battery will run for around 2 hrs. We use a VGA camera with 2 megapixels.

III. PROBLEM STATEMENT

Our objective is to design and construct a robot to be used in an emergency response mission that will be able to venture into terrain and environments that are otherwise too dangerous for human responders. The main function of the Rescue Robot is to detect survivors using real time video transmission and send robot location via Bluetooth GPS receiver mounted on robot. The video camera will also be connected to PC wirelessly for real time video transmitting. The GPS will also send location data to PC and it will have the location where rescuers are to be sent. The robot is controlled by Microcontroller and will be driven via looking at real-time video feedback received on

PC. Intelligent mobile robots and cooperative multi-agent robotic systems can be very efficient tools to speed up search and rescue operations.

IV. DESIGN

A. ARCHITECTURE:

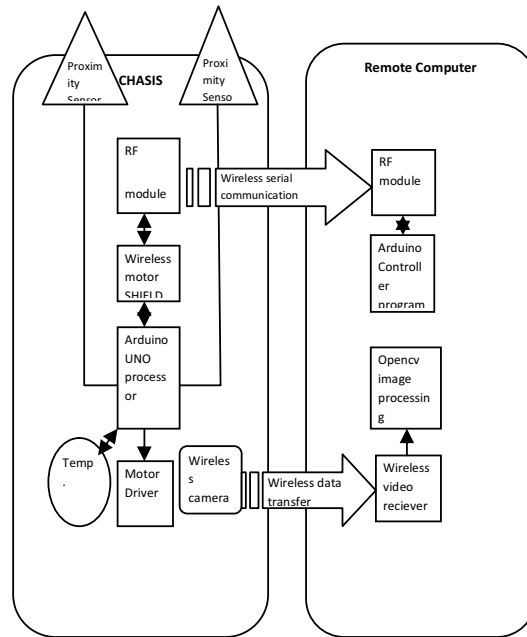


Fig.1 Architecture Diagram

The architecture diagram shows how, the robot is connected to the remote computer. It shows the internal architecture of our project. The arduino ^[6] processor will be connected to the motor driver responsible for the movement of the robot. The RFpro will be connected to the Arduino shield which will be connected to the Arduino processor. The whole chassis will be connected to the PC through the RF pro which is received by the GUI created on the PC. The GUI is responsible for giving direction to the robot. The camera will be providing video streaming to the remote controller which will process the image being received by Opencv Visual studio C++ codes. The remote computer will also have a RFpro which will be wirelessly connected to the RFpro in the chassis. The video received by the PC through the RF pro is then processed to show the faces of people trapped in the rubble.

VI.HARDWARE AND SOFTWARE REQUIREMENTS

A. Hardware requirements:

- Arduino SD Shield
- Arduino Proximity sensors with ultrasonic range finder
- A light sensor

B. Software requirements:

- Opencv
- Visual studio

Hardware requirements:

Sr no	Name of the Hardware	Features
1	Arduino UNO	<ul style="list-style-type: none"> • Microcontroller ATmega328 • Operating Voltage 5V • Input Voltage (recommended) 7-12V • Input Voltage (limits) 6-20V • Digital I/O Pins 14 (of which 6 provide PWM output) • Analog Input Pins 6 • DC Current per I/O Pin 40 mA • DC Current for 3.3V Pin 50 mA • Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader • SRAM 2 KB (ATmega328) • EEPROM 1 KB (ATmega328)
2	RF pro	<ul style="list-style-type: none"> • Baud Rate: 9600 to 38400. Serial UART Mode • Packet Length: Variable or Fixed . • 30+ meters range Line of Sight / 10 meters range indoors • Multiple channel selection enabling upto 255 different pairs to work in the same area • Modes of operation: Config mode and Run mode. • Onboard jumper Setting for Config/Run Mode and Packet/Byte Mode. • Direct Replacement for wired Serial Cable for and serial communication • Working with CC2500 Wireless Transceiver module
3	Fire detector sensor	<ul style="list-style-type: none"> • Allows robot to detect flames from upto 2m away • Typical Maximum Range : 2 m . • Calibration preset for range adjustment. • Indicator LED with 3 pin easy interface connector.
4	52 chain links	For a belt drive mechanism one can directly make use of 52 chain links together to have two belts of 26 links each.
5	Wheels	The diameter of this wheel is 85 mm and the width is 12.5 mm.
6	Battery	<ul style="list-style-type: none"> • Output Voltage: 12V. • 2.Output Current: 1300mAH.
7	Multimeter	<ul style="list-style-type: none"> • DC Voltage - 200mV/2V/20V/200V/600V • AC Voltage - 200V/600V • DC Current - 200uA/20mA/200mA/10A • Resistance - 0 to 20M Ohm • Continuity Buzzer • Diode Test

		<ul style="list-style-type: none"> • Max display: 1999
8	iBall Claro TV18 TV Tuner Card	<ul style="list-style-type: none"> • External TV Tuner (TV Tuner Stick) for PC, Laptop • Remote Control Supported, Remote Control Included • FM Radio with Recording • Video Analog-to-Digital Converter • TV Recording in MPEG-1, MPEG-2, MPEG-4, AVI • Image Capture Support • S-video in, USB

These are the various components connected:



Fig 2. Internal connections of the robot

V. IMPLEMENTATION

A. CROSS PLATFORM CONNECTION

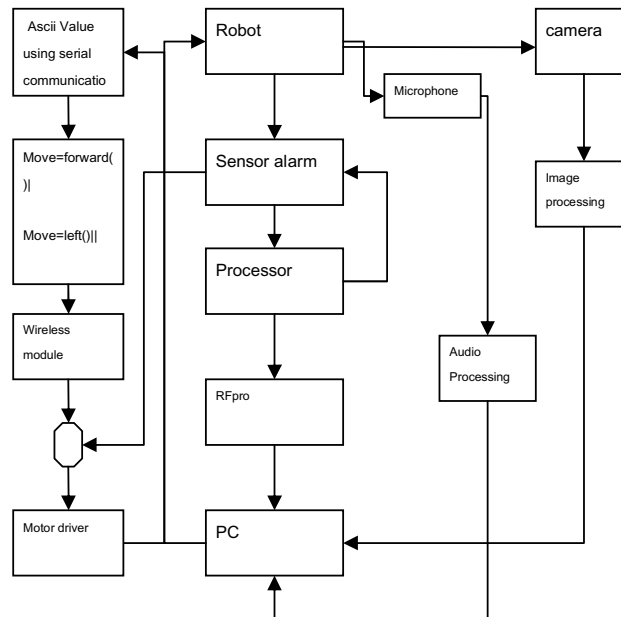


Fig 3. Flow diagram of rescue robot

B. TRANSMISSION FLOW

The robot will be started and checked if it is working properly and if it is not working, we will check what the problem is and make changes accordingly. The actual working inside the robot is as follows: We have the GUI connected to the HyperTerminal. The HyperTerminal sends value through serial communication to the COM ports in which communications happen via USB. The USB output is converted to serial output using USB to serial convertor. The output of the serial port consists of 9 pins, but the input of RFpro consists of only 3 pins namely, RX, TX and GND. For this we use 9 pin to 3 pin convertor and it is connected to Rf pro. At the same time the RFpro module is also interfaced to the arduino robot shield which is interfaced to arduino UNO board where the program is preloaded. Based on the values sent through the hyperterminal the switch cases within the program selects the particular action. In the mean while there are sensors which are interfaced to the arduino robot shield which senses for its particular application and will send an output to the UNO board. Based on these outputs, the consequent actions will be chosen and in some cases relay message will be flashed on the PC.

C. CONTROLS FOR ROBOT MOVEMENT

In our program, when command is given through the GUI, the Ascii ^[7] keys A, W, S, D and X are activated. It is used to move left, forward, backward, right and stop respectively.

D. VIDEO DATA TRANSMISSION

When our robot moves through the rubbles, it provides live streaming of the area. For this purpose, we have used wireless security camera which is available cheap in the market and operates on radio frequency. It is also equipped with a microphone, so audio and video can be sent to the PC simultaneously for the further processing. Our face detector would track the faces of people trapped under the rubbles.

Face detection is finding whether or not there are any faces in a given image and, if present, return the image location and content of each face. Face Detection algorithms can be implemented using digital image processing techniques such as edge detection, eye detection.

E. FACE DETECTION ALGORITHMS

Numerous representations have been used for face detection algorithms, including pixel-based, parts-based, local edge features, Haar wavelets and Haar-like features.

Performance of **face detection algorithms** can be evaluated using the following terminology:

- Detection rate: The number of faces correctly detected to the number of faces determined by a human expert (hand-segmented results).
- False positives: This is when an image region is declared to be a face but it is not.
- False negatives: This is when an image region that is a face is not detected at all.
- False detections: False detections = False positives + False negatives.

When our robot moves through the rubbles, it provides live streaming of the area. Our face detector would track the faces of people trapped under the rubbles.

Face detection can be done using MATLAB also. But the disadvantage in using MATLAB is:

It detects faces statically and not dynamically.

It is more costly.

Face detection using skin detection ^[12]:

As was said earlier, there are different methods for face detection. It can be done using skin detection. Skin detection is usually used as a preprocessing step for the more advanced image analysis, like face detection and recognition.

//Step 1 - Load the image

Load the source image and make sure the it is successfully loaded.

```
CvCapture* capture=cvCreateCameraCapture(1);
```

```
IplImage *fram=cvQueryFrame(capture);  
if(!fram)  
break;
```

```
//Step 2 - Convert to HSV  
Convert the loaded image to HSV colorspace.
```

```
cvCvtColor(frame,hsvframe,CV_BGR2HSV);
```

```
//Step 3 - Threshold the image
```

Given the HSV image above, threshold the image so the pixels where the HSV values match with human skin tone are set to 255 and 0 otherwise. To define human skin tone, we set an upper and lower boundary in HSV colorspace.

```
cvInRangeS(hsvframe,cvScalar(h1,s1,v1),cvScalar(h2,s2,v2),threshy);
```



Fig 4. Output of Skin detection

Disadvantage of using skin detection is:

It depends in light conditions and hence needs to be adjusted according to the lighting conditions.

Face detection using Viola- Jones^[11] method

The face detector of this project is realized in OPENCV, based on Viola-Jones' approach. Our face detection system classifies images basing on the value of four particular kind of Haar-like features, so the first step is the implementation of a function capable to generate randomly sets of features. The second step is the realization of the boosting algorithm, the procedure that, given a set of simple weak classifiers (each one with a low accuracy) elaborated on a training set of positive and negative face examples, round by round builds a strong classifier with high accuracy by the combination of the T best weak classifiers. The third step is the realization of the cascade, an optimization process designed for reducing the number of features to use and so useful to increase the general performances. The last step consists in testing the system on a set of examples and observing his behaviour, in terms of detection rate and accuracy.

Viola-Jones' method is based on the notion of feature; a feature corresponds to a set of two or more adjacent rectangles. The value of a feature is computed by subtracting the sum of the pixels within the white rectangle from the sum of the pixels within the black one.

This function is present in Opencv in the cascade classifiers.

After a classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image. The classifier outputs a "1" if the region is likely to show the object (i.e., face/car), and "0" otherwise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily "resized" in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales.

To classify an image, we use a set of 3 different kind of features:

2 rectangle feature

3 rectangle feature

4 rectangle feature

The word "cascade" in the classifier name means that the resultant classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word "boosted" means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques (weighted voting). Currently Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost are supported. The basic classifiers are decision-tree classifiers with at least 2 leaves. Haar-like features are the input to the basic classifiers, and are calculated as described below. The current algorithm uses the following Haar-like features:

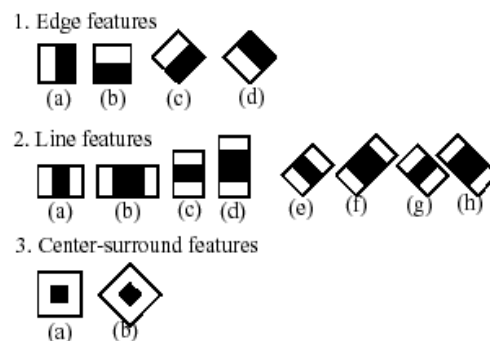


Fig 5. Haar features

The feature used in a particular classifier is specified by its shape (1a, 2b etc.), position within the region of interest and the scale (this scale is not the same as the scale used at the detection stage, though these two scales are multiplied). For example, in the case of the third line feature (2c) the response is calculated as the difference between the sum of image pixels under the rectangle covering the whole feature (including the two white stripes and the black stripe in the middle) and the sum of the image pixels under the black stripe multiplied by 3 in order to compensate for the differences in the size of areas. The sums of pixel values over a rectangular region is calculated rapidly using integral images.

Face detector steps can be summarized as follows:

//Step 1: Load video

```
CvCapture* capture=cvCreateCameraCapture(1);
```

//Step 2: Initialize detector

```
CascadeClassifier cascade;
```

//Step 3: Running the detector

```
Mat gray;
cvtColor(frame, gray, CV_BGR2GRAY);
equalizeHist(gray, gray);

vector<Rect> faces;
cascade.detectMultiScale(gray, faces, 1.2, 3);
```

//Step 4: Detector Results

```
imshow("src", frame);
```

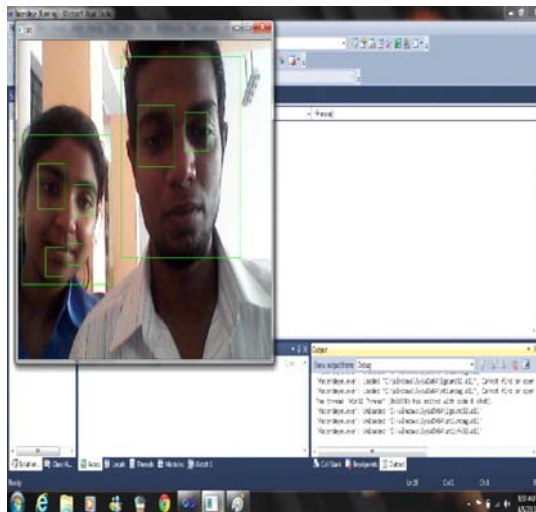


Fig 6. Output of face detection

F. TESTING AND DEBUGGING

The box strip of the arduino robot shield had to be deshouldered and to be shouldered back with a longer box strip as it was difficult to interface arduino motor shield to the arduino UNO board directly as due the presence of ICs. While doing so, the track of the PCB was at risk of being damaged. To overcome this problem, we took help of a professional electronics engineers.

Then, the battery went down and as a result the robot would not move. Hence we had to use a spare battery and recharge the previous one.

The robot was not able to overcome small obstacles and so we added the gripper to the track tires and then the robot was able to cover the obstacles.

The other problem we faced was, while installing Opencv. It was installed easily in 32-bit processors but not on 64-bit machines. So we went through many forums and then cleared the installation.

While running the program, it threw memory heap error after detecting a single frame. After researching we came to a know that, the error was a result of using unlicensed version of visual studio. After using a licensed version, the error was no more and our face detection worked fine.

VI. FUTURE DEVELOPMENT

With the current scenario, we will add thermo sensors to our robot which will enhance the face detection process. The thermo sensors will enable to find if there is life inside the rubbles. We will also add additional arms to the robot which will help our robot to clamber particular heights. We will make this robot water proof upto 30 m. we can also use Xbee wi fi shield instead of RF pro module which has better range as compared to RF pro. Its range would depend on the number of routers used and placed strategically.

REFERENCES

- [1] "Shahram Bahadori, Daniele Calisi, Andrea Censi, Alessandro Farinelli, Giorgio Grisetti, Luca Iocchi, Daniele Nardi", "Intelligent Systems for Search and Rescue", <http://www.dis.uniroma1.it/~calisi/papers/iros04rescue.pdf>, 15th july 2012
- [2] "Amon Tunwannarux and Supanunt Hirunyaphisuthikul", "Design Features and Characteristics of a Rescue Robot", "http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1567056&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D1567056", 15th July 2012
- [3] "Mahmud, Faisal. ; Hossain, S.G.M. Alam, Jobair Bin.", "Low- Cost Rescue Robot for Disaster Management In a Developing Country - Development of a Prototype Using Locally Available Technology", "http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100012838_2010013718.pdf", 22nd July 2012
- [4] "Matteo Venanzi, Leonardo Benucci", "Face Detection", "<http://www.dis.uniroma1.it/~visiope/Tesine2008/BenucciVenanzi/Documentation.pdf>", 22nd July 2012
- [5] <http://www.allonrobots.com/rescue-robots.html>
- [6] <http://arduino.cc/en/Tutorial/HomePage>
- [7] <http://arduino.cc/en/Tutorial/ASCIITable>
- [8] <http://arduino.cc/en/Reference/HomePage>
- [9] "Fundamentals of Robotics" – "PHI learning pvt. Ltd.", Robert J. Schilling
- [10] "Jing Chen, Shaoming Chen", Face Tracking, "<http://www.ece.lsu.edu/ip1/SampleStudentProjects/ProjectChen/report%20of%20face%20tracking.pdf>
- [11] "Carol Reiley and Daniel Mirotu", face detection, <https://www.lcsr.jhu.edu/main/images/9/9a/EN600.406Lecture5.pdf>
- [12] "Yao-Jiunn Chen, Yen-Chun Lin", Simple Face-detection Algorithm Based on Minimum Facial Features, "<http://wenku.baidu.com/view/bcf2363143323968011c9200.html>