C-Dog:

Computer-based Direct Object Guidance

Assisting the Visually Impaired Using

Object Recognition

By

Paul Steven Weiss

Fox Lane High School, 2007/2008

Advisor: Nancy Williams
Introduction

Computer Vision with Object Recognition is an innovative concept in the world of technology and computing. There are many paths to follow when dealing with Computer Vision; medical, as in tele-surgery, industrial, for assembly lines, military, for guided missiles and other guided artillery, and in robotics. Robotics and military uses are advancing and evolving very quickly. A lot of research is being done at labs and universities around the country and the world. The Defense Advanced Research Projects Agency as well as New York University, Massachusetts Institute of Technology, California Institute of Technology, Carnegie Mellon University, and Stanford University are just some of the front runners in advancing CV technology for a new age.

Goal of Project

The goal of C-Dog is as such; design, create, test, and implicate computer vision and object recognition so that people who are visually impaired have a way to navigate a given situation. This project should, in the long term, enhance the lifestyle of those who are visually impaired because it can give them better mobility and safety when walking around.

Background

Navigation using Computer Vision is currently been used in the expanding field of “driverless cars.” The Defense Advanced Research Project Agency (DARPA) Grand Challenge posed a new question to the scientific community; how can we make cars go on their own? In 2005, 23 teams of scientist, students, teachers, and corporations
answered the call. Entrants like “Stanley”, the Stanford University team car, “TerraMax”, a 30,000 pound Oshkosh truck, and Carnegie Mellon University’s “Sandstorm” and “H1ghlander” used large amounts of new technology to complete the race. Teams used Computer Vision to assist the cars when avoiding objects such as rocks, boulders, and bushes. This idea of object avoidance using Computer Vision can be applied to human-based navigation.

The robot that won the DARPA Grand Challenge in 2005, “Stanley”, used an array of sensors and cameras to help itself navigate (Stanford Racing Team, 2005). “Stanley” used a system in which a forward looking camera observed the road conditions in front of the car. The system also utilized a LIDAR, Light Detection and Ranging, that, when fused with the video components, found safe passages to drive on (Figure 1). The system could then make decisions also on how fast to go in certain areas; if a passage was deemed unsafe, the car would slow down as to not hinder itself if a problem or obstacle arose (Stanford Racing Team, 2005).
Racing Team, 2005). This system can be compared to the system of C-Dog, in that it uses its forward looking camera to find obstacles and problems as they arise.

At the Georgia Institute of Technology, Professors Bruce Walker PhD., and Frank Dellaert PhD created the “SWAN”, System for Wearable Audio Navigation. Their system uses a series of cameras, sensors, GPS readouts, and compasses to help blind people get from one preset point to another. SWAN does not explicitly use cameras for navigation; it relies heavily on the use of GPS to help the user stay on a course. The system also uses a compass-like device, mounted on the top of the head, which has the ability to detect which way the user is facing as well as the degree of the incline or decline of the head, thus directing the person in the correct way. This system can be compared to an in-car navigation system. SWAN also uses a device called “bonephones.” (Figure 2)

“Bonephones” is a device that acts as headphones for SWAN, but it does not go over the ears, instead, the speaker is located behind the ear and emits the sound into the bone around the ear which is then received by the cochlea. They emit a beacon-like sound that notifies the user if they are getting closer to the target waypoint by increasing the speed of the beacon (Walker, 2006).
SWAN differs from C-Dog because SWAN is for people who are completely blind, whereas C-Dog is for people who are not completely blind so they need help doing everyday activities.

Dr. Yann LeCun at New York University’s Courant Institute of Mathematical Sciences works extensively with Computer Vision and Machine Learning. One of the projects directly relating to object recognition is the NORB project, the NYU Object Recognition Benchmark. NORB is a project designed to specifically target the idea of recognition purely based on the shape of the object independent of color, background, and surrounding visual clutter.

**Methods and Materials**

C-Dog utilizes the capabilities of a Logitech QuickCam Webcam (Figure 3). The QuickCam is connected, via USB, to an IBM ThinkPad T23 (Figure 4) running Ubuntu as an operating system. C-Dog also uses OpenCV (CV stands for...
Computer Vision), an open-source library developed by the Intel Corporation. Since OpenCV is used commonly for real-time image processing, it was ideal for this project.

There are a few steps that needed to be taken in order to train the system. The first step was to go out, in this case to New York City, and take pictures of the crosswalks (Figure 5). After collecting enough samples, a Haar Cascade Classifier was run “over” the pictures. The program finds certain interest points in the picture. This is what the program compares the other pictures to. This process makes the pictures, and the real-time data, comparable and recognizable. The Haar Cascade Classifier has the ability to store and access the comparable data. The classifiers are accessed in real-time when C-Dog is activated. The classifier will compare the given real-time image from the camera attached to the system. The C-Dog algorithm then processes the data. C-Dog then makes a decision to tell or not to tell the user of what is seen based on the facts of the scenario.

OpenCV provides a face recognition algorithm for people to use (Bradski, 2005). After several modifications to the original algorithm, the program was able to recognize faces and profiles, the side of the head. This was then incorporated into the C-Dog project as an added feature which could be useful if someone can not see people clearly. C-Dog can recognize faces and profiles.
The code for C-Dog is in three parts; LEASH, the project file, and the festival client. The project file runs the recognition process and classifies any items put into the program. The main code for this part is displayed later in this essay. LEASH is the learning environment for C-Dog. It also communicates with the festival client. LEASH stands for Learning Environment and Speech Host, the acronym goes along with the idea of dogs, as in C-Dog. Festival is the speech synthesis program developed by the Centre for Speech Technology Research at the University of Edinburgh in the United Kingdom. The client is used to deliver the gathered information from C-Dog to the user via auditory messages. As of now, auditory messages will be used until another viable option is found. The auditory message for crosswalks is “Crosswalk detected,” for faces it is “(number of faces) face(s) detected,” for profiles it is “(number of profiles) profile(s) detected.”

The original plans for the project called for the use of SIFT (Scale-Invariant Feature Transform) as a classifier instead of Haar Cascade Classifiers. Both Haar and SIFT have their pros and cons. SIFT is flexible, meaning it is invariant to scale, shading, rotation, and angle when recognizing an object and requires less training compared to Haar. The underlying problem is that SIFT is patented by David Lowe at the University of British Columbia. There was also the matter of SIFT being extremely hard to understand and implement correctly because of the enormities and the complexities of the code itself. Haar is comparatively faster than SIFT in recognizing objects and running real-time data. It is also much easier to work with in terms of implementation and management. The draw backs are that it takes an enormously large training sample and it has a low tolerance for angles and positions. Nevertheless, Haar had to be used because
of the patent issue. There are a few other kinds of object recognition algorithms such as ShapeContext, developed by University of California at Berkeley, and NORB, developed at New York University. ShapeContext requires an enormously large sample data set, 10,000 to 20,000 samples are recommended (Belongie 2001). NORB, although developed in the lab where this project took place, was not available because it did not work on a real-time basis.

**Testing and Results**

To test C-Dog, the Haar Classifier was run over various positive and negative pictures. Positive pictures are ones that contain crosswalks in them. Some positives were gathered in the Greenwich Village section of New York City, Greenwich Village is where New York University is located. More positives and negatives were gathered from online resources such as MIT’s LabelMe annotation dataset. In total, about 1,400 useable pictures were gathered. Once the test set of pictures were imputed into the classifier, testing could start.

Since OpenCV comes with a face recognition utility. C-Dog utilized the face recognition utility in testing. C-Dog was enhanced to include profiles of faces along with strait on faces. OpenCV’s face recognition has been independently tested by Intel.

**Results**

C-Dog can not be tested as originally anticipated. Yet, as an object recognition machine, C-Dog worked as planned. It successfully recognized images of faces and profiles in all three tests.
The following is most of the code for C-Dog. This is the main code section for the system. Omitted here are some #include lines as well as various comments and notations that are not necessary and/or extraneous.

```c
//CDOG: Computer-based Direct Object Guidance
//2007 Paul Weiss
//File: project.c
//Description: main code file

#define GC_DEBUG
#include "gc.h"
#define CHECK_LEAKS() GC_gcollect()
#include "cv.h"
#include "highgui.h"
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <float.h>
#include <limits.h>
#include <ctype.h>
#endif

static CvMemStorage* storage = 0;
int detect_and_draw( IplImage* image);
void display_num_objects(int num, const char name[]);
static int numobjects;
typedef struct{
    CvHaarClassifierCascade* cascade;
    char name[50];
} Recognizable;
Recognizable* objects;

int main(int argc, char** argv){
    GC_init();
    cvSetMemoryManager(((CvAllocFunc)GC_debug_malloc_uncollectable), ((CvFreeFunc)GC_debug_free), NULL);
    CvCapture* capture = 0;
    IplImage *frame, *frame_copy = 0;
    const char* input_name;
    FILE* list = fopen("objectlist", "rt" );
    if(!list){
        fprintf(stderr, "Error: couldn't open the object list\n");
```
return -1;
}
char buf[1000];
fgets(buf, 1000, list);
numobjects = atoi(buf);
objects = GC_MALLOC((sizeof (Recognizable))*numobjects);
Recognizable* tempref = objects;
int valid = 0;
for(int i=0; fgets(buf, 1000, list) && (i < numobjects); i++){
    tempref->cascade = (CvHaarClassifierCascade*)cvLoad(strtok(buf, " "), 0, 0, 0);
    strcpy(tempref->name, strtok(NULL, "\n"));
    tempref++;
    valid++;
}
fclose(list);
if(valid < numobjects) objects = GC_REALLOC(objects, (sizeof (Recognizable))*valid);
printf("Loaded %d objects\n", valid);
printf("Heap size = %d", GC_get_heap_size());
if(valid == 0){
    fprintf(stderr, "Error: couldn't load any valid objects\n");
    return -1;
}
numobjects = valid;
storage = cvCreateMemStorage(0);
capture = cvCreateCameraCapture(200);
if(!capture){
    fprintf(stderr, "Error: couldn't open camera for capture\n");
    return -1;
}
cvNamedWindow( "View", 1 );
for(;;){
    frame = cvQueryFrame( capture );
    if( !frame ) break;
    if( !frame_copy )
        frame_copy = cvCreateImage( cvSize(frame->width,frame->height),
                                   IPL_DEPTH_8U,
                                   frame->nChannels );
    if( frame->origin == IPL_ORIGIN_TL )
        cvCopy( frame, frame_copy, 0 );
    else
        cvFlip( frame, frame_copy, 0 );
    int untilquit = detect_and_draw(frame_copy);
    if(untilquit == 27) break;
printf("loop cycle\n");
cvWaitKey(10);
CHECK_LEAKS();
}

for(int i=0; i<valid; i++){
    cvReleaseHaarClassifierCascade(&objects->cascade);
    objects++;
}

cvReleaseMemStorage(&storage);
cvReleaseImage( &frame_copy );
cvReleaseCapture( &capture );
cvDestroyWindow("View");
CHECK_LEAKS();
return 0;
}

int detect_and_draw(IplImage* img){
    static CvScalar colors[] =
    {{0,0,255}},
    {{0,128,255}},
    {{0,255,255}},
    {{0,255,0}},
    {{255,128,0}},
    {{255,255,0}},
    {{255,0,0}},
    {{255,0,255}}
};

double scale = 1.4;
IplImage* gray = cvCreateImage( cvSize(img->width,img->height), 8, 1 );
IplImage* small_img = cvCreateImage( cvSize( cvRound (img->width/scale),
cvRound (img->height/scale)),
8, 1 );

int i;

cvCvtColor( img, gray, CV_BGR2GRAY );
cvResize( gray, small_img, CV_INTER_LINEAR );
cvReleaseImage(&gray);
cvEqualizeHist( small_img, small_img );

Recognizable* tempref = objects;
for(int j=0; j<numobjects; j++){
    int numdetected = 0;
    double t = (double)cvGetTickCount();
    CvSeq* results = cvHaarDetectObjects(small_img, tempref->cascade, storage,
    1.4, 2,
CV_HAAR_DO_CANNY_PRUNING,
    cvSize(100, 100));
    t = (double)cvGetTickCount() - t;
    printf("detection time = %gms\n", t/((double)cvGetTickFrequency()*1000.));
}
C-Dog is important because it is a new and innovative way of self mobility in this fast paced world that we live in. C-Dog will be, in a sense, a Seeing Eye Dog. This system could be the future of domestic personal help animals.

Besides the main focus of C-Dog, aiding those who are visually impaired, there are many foreseeable implementations; the most distinguished of which is use in the armed forces. C-Dog could be used instead of a soldier to go into buildings or other fortitudes and seek out prospective targets predefined by a ranking official. Since this program is adaptable to many situations, it could be built into a mobile robot controlled autonomously or by a soldier. In addition, the program could have the ability to control itself using simple autonomous commands written into the program. In the larger picture, the program could execute, incorporated with a mobile robot, and find all of the targeted
items over a large area, thus eliminating the need for soldiers to scout the area and potentially saving many lives due to explosive devices undetected pre-operation. In continuance, the camera and, subsequently, the program could execute with an image intensifying or night vision device attached to the camera. Since the relative shape and complexities of the object are preserved, C-Dog would still have the ability to recognize the object, theoretically.

Along similar parameters as a military use, governmental and private investigative intelligence agencies could use C-Dog in a similar manner. Based on the object recognition capabilities, intelligence gathering entities could use C-Dog to find objects without using a human, thus eliminating the unnecessary danger put on a human to do a job that C-Dog could do. Once trained on an object, C-Dog would easily find that given object in a given situation.

Another main topic of study in future research would be using solar energy to power the C-Dog system. This way, there would be extended battery life and it would make C-Dog a “green solution.”

The main underlying problem with C-Dog and similar systems is the inability to recognize every object that one would encounter in the world. This is because of the large amount of time needed to implement thousands of objects. It is a problem that will persist throughout the course of the Computer Era because everything that should be done in order to perfect a system, such as C-Dog, takes time and time waits for no one.

**Conclusion**
C-Dog, with more research and time spent on perfecting the recognition, classification, and notification algorithms, could be a viable product for people affected by visual impairment.

Nevertheless, C-Dog, in its current state, works as anticipated in that it can successfully recognize given objects, such as the crosswalk, faces, and profiles. By this fact alone, it can be interpreted that C-Dog could recognize many objects once implemented.

The concept of C-Dog is the most important part of the project. Hopefully, this will be further researched and developed into a useable product for the modern era in as many ways as possible.

**Future Outlook**

One day, hopefully, a time when mankind and robots interact seamlessly will arise. It is possible that this project could be one of the many stepping stones that era. It is important to embrace this possibility because robots and computers can give us so much; for example, accessing information rapidly, helping us drive, doing surgery to save our life, and even helping us see when we can’t on our own.
Cited Resources

http://citeseer.ist.psu.edu/article/belongie02shape.html

http://developer.intel.com/technology/itj/2005/volume09issue02/art03_learning_vision/p01_abstract.htm

