Contact Detection and Analysis System (CDAS)

Maritime Autonomy and Perception

Helena Rhee

Work supported by National Science Foundation (NSF)
Grant #14060538 to Los Angeles City College, CURE program

Background

- System development
  - Unmanned Surface Vehicles (USV)
    - Navigate
    - International Regulations for Preventing Collision at Sea (COLREGS)
  - CDAS
    - Runs onboard
    - Support of motion planning decisions for robotic boat
- Mission operations
  - Automated Target Recognition
  - Intelligence, Surveillance, and Reconnaissance (ISR)

Work supported by NSF Grant #1460538

Jet Propulsion Laboratory
California Institute of Technology

Process

1. CDAS captures images from Hammerhead systems
2. Images are stored on a disk
3. Disk is sent to JPL

Work supported by NSF Grant #1460538

Jet Propulsion Laboratory
California Institute of Technology

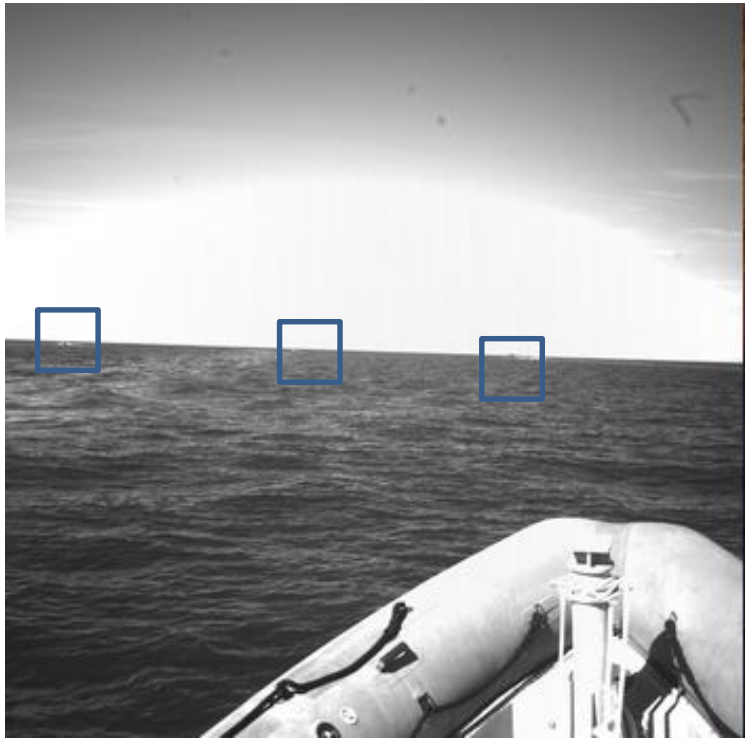
Annotations

- Bound objects found in the image
- Label the bound box

pleasure

military:small

military:small



Work supported by NSF Grant #1460538

Jet Propulsion Laboratory
California Institute of Technology

```
camera1-0-00797.pgm
#ownship -999      -999     -999     -999     -999     -999
#num_contacts 2
#contact 41 pleasure  130.4 1191.0  101.4   38.5     -999
#contact 42 military:small 2679.2 1342.3   60.5   34.1     -
```

- Annotation File
  - Vessel types
  - Bounding box values
- Image File

LACC LOS ANGELES CITY COLLEGE
Work supported by NSF Grant #1460538
Jet Propulsion Laboratory
California Institute of Technology

Bash Script

- Data Validation
- Routing to Directory
- Search through Annotation Files
- Extract the bounding box values
- Produce Image Chips

```
#####Functions
dataVal() #data input validation
{
        if [[ -z $dirname || -z $outDirPath || -z $vesselType ]]; then
                echo
                echo "Missing a command line argument"
                exit
        elif [ ! -d "$dirname" ]; then
                echo
                echo "Base Directory does not exist"
                exit
        elif [ -z $optPad ]; then
                optPad=0
                extAmt=-1
                extHe=-1
        elif [ -z $extAmt ]; then
                extAmt=-1
                extHe=-1
        elif [ -z $extHe ]; then
                extHe=$extAmt
        fi #ends if for empty arguments
} #ends dataVal
routeDirname() #route to directory-doesn't change the terminal directory, create output directory & add txt file
{
        test -d "${outDirPath}" || mkdir -p "${outDirPath}"
        echo "the output directory path is: $outDirPath"
        cd $dirname
} #ends routeDirname
searchType() #searches for the vesselType and stores into temporary file
{       echo
        echo "Searching..."
        searchExp=$(find|grep ant| xargs grep "#contact"| cut -d ' ' -f3|grep -Fx $vesselType|sort -n|uniq -c)
        echo "$searchExp instance(s) were found"

        #Tempfile to hold the results of the search
        TMPFILE=`mktemp`

        #Search all the ant files for the vesselType
        find|grep ant| xargs grep "#contact"| egrep "[[:space:]]$vesselType[[:space:]]"  > $TMPFILE
        while IFS= read -r line; do
                fullarray+=("$line")
        done < "$TMPFILE"
        #cleans the file
        truncate -s 0 $TMPFILE

        for (( q=0; q < ${#fullarray[@]}; q++)); do
                fname+=("$(echo ${fullarray["$q"]}| awk -F: '{print $1}')")
        done
        fname=(`for i in ${fname[@]}; do echo $i; done |sort -u`)

        #Stores the ant and pgm files to tempfile
        for t in ${fname[@]}
        do
                p1=$(dirname $t)
                p2=$(basename $t)
                p=$(basename $p1)

                #strips the path to the last two, excludes the ext
                p3="${p2%%.*}"
                p4=$(basename $p1)/$p3

                #for the pgm file
                p5="$p1/$p3.png"   ######------------------> change this part for ext##################

                #appends to the tempfile
                echo "$t" >> $TMPFILE
                echo "$p5" >> $TMPFILE

                echo "the value of ant going into tmpfile is: $t"
                echo "the value of png going into tmpfile is: $p5"
        done #ends the for statement- list of ant files
 } #ends searchType
```

LACC LOS ANGELES CITY COLLEGE

Work supported by NSF Grant #1460538

Jet Propulsion Laboratory
California Institute of Technology

```
processFiles() #process the files in the temporary file
{
        #read the tempfiles and store them in two separate arrays respective to the ext
        while IFS= read -r line; do
                if [[ "$line" =~ '.ant' ]]; then
                        antarray+=("$line")
                else
                        pgmarray+=("$line")
                fi
        done < "$TMPFILE"

        #processes the ant and pgm using two arrays
        for (( q=0; q < ${#antarray[@]}; q++ ));do
                echo "the value of antarray: ${antarray["$q"]}"
                output=$(awk '/'$vesselType'/ {print $2,$4,$5,$6,$7}' ${antarray["$q"]})
#               echo "the value of output: $output"
                IFS=$'\n' read -d '' -r -a output1 <<< "$output"
                        for h in "${output1[@]}"; do
                                array=($h)
                                echo "the length of output1 array ${#output1[@]}"
                                echo "the value in the array, separate the output array: ${array[@]}"
                                contact=${array[0]}
                                xmin=${array[1]}
                                ymin=${array[2]}
                                width=${array[3]}
                                height=${array[4]}

                                #use size variables and offset values to crop images
                                XSIZE=`identify -format '%w' ${pgmarray["$q"]}`
                                YSIZE=`identify -format '%h' ${pgmarray["$q"]}`
                                count=$((count+1))
                                echo "Processing $count file"
                                echo "value of contact: $contact"
                                echo "value of xmin: $xmin"
                                echo "value of ymin: $ymin"
                                echo "value of width: $width"
                                echo "value of height: $height"
                                echo "value of XSIZE: $XSIZE"
                                echo "value of YSIZE: $YSIZE"
                                echo

                                #calculate center pixels of bounded image
                                cenx=$(echo "((2*($xmin)+$width))/2" | bc -l)
                                ceny=$(echo "((2*($ymin)+$height))/2" | bc -l)
                                echo "the coordinates of the center pixels ($cenx,$ceny)"

                                 #calculate the left, right, top, bottom to test boundaries
                                 #when extended values are given
                                 if [[ "$(echo "$extAmt" | bc)" != "-1" ]]; then
                                        extAmt=$(echo "$extAmt/2" | bc -l)
                                        extHe=$(echo "$extHe/2" | bc -l)
                                        lmxchg=$(echo "$cenx-($width/2)"| bc -l)
                                        rmxchg=$(echo "$XSIZE-($cenx+($width/2))"| bc -l)
                                        tmychg=$(echo "$ceny-($height/2)"| bc -l)
                                        bmychg=$(echo "$YSIZE-($ceny+($height/2))"| bc -l)
                                        xchg=$(echo "$cenx-($extAmt+($width/2))"| bc -l)
                                        ychg=$(echo "$ceny-($extHe+($height/2))"| bc -l)

                                        echo "the value of extAmt: $extAmt"
                                        echo "the value of extHe: $extHe"
                                        echo "the value of lmxchg: $lmxchg"
                                        echo "the value of rmxchg: $rmxchg"
                                        echo "the value of tmychg: $tmychg"
                                        echo "the value of bmychg: $bmychg"
                                        echo "the value of xchg: $xchg"
                                        echo "the value of ychg: $ychg"

                                        #test which sides are greater
                                        if (( $(echo $lmxchg '<=' $rmxchg|bc -l) )); then
                                                xmargin=$lmxchg
                                        else
                                                xmargin=$(echo "$lmxchg-$rmxchg"| bc -l)
                                        fi #ends xmargin
                                        if (( $(echo $tmychg '<=' $bmychg|bc -l) )); then
                                                ymargin=$tmychg
                                        else
                                                ymargin=$(echo "$tmychg-$bmychg"| bc -l)
                                        fi #ends ymargin

                                        echo "the value of xmargin: $xmargin"
                                        echo "the value of ymargin: $ymargin"

                                        #test if exceeds image bounds
                                        if (( $(echo $xchg '<' "0"|bc -l) )); then
                                                echo "xchg < 0 if statement executes"
                                                xchg=0
                                                extAmt=$xmargin
                                                echo "value of xchg changed to: $xchg"
                                                echo "value of extAmt changed to: $extAmt"
                                        fi #ends xchg=0
                                        if (( $(echo $ychg '<' "0"|bc -l)) ); then
                                                ychg=$0
                                                extHe=$ymargin
                                        fi #ends ychg=0

                                        #resize and rescale image
                                        xmin=$(echo "$XSIZE-$xchg"|bc -l)
                                        cenx=$(echo "$cenx-$xchg"|bc -l)
                                        ymin=$(echo "$YSIZE-$ychg"| bc -l)
                                        ceny=$(echo "$ceny-$ychg" | bc -l)

                                        echo "the rescale values are"
                                        echo "xmin is: $xmin"
                                        echo "ymin is: $ymin"
                                        echo "the new center is (cenx, ceny): ($cenx , $ceny)"

                                        #check offsets
                                        xoffset=$(echo "$xmin -($cenx+($width/2)+$extAmt)"|bc -l)
                                        yoffset=$(echo "$ymin-($ceny+($height/2)+$extHe)"|bc -l)
                                        echo "the xoffset value is: $xoffset"
                                        echo "the yoffset value is: $yoffset"
                                else
                                        #hits when no extended values have been given
                                        xchg=$xmin
                                        ychg=$ymin
                                        xmin=$(echo "$XSIZE-$xchg"|bc)
                                        ymin=$(echo "$YSIZE-$ychg"|bc)
                                        xoffset=$(echo "$xmin-$width"|bc)
                                        yoffset=$(echo "$ymin-$height"|bc)
                                fi  #ends $extAmt != -1 statement


                                #Adds modifier
                                modifier=$(echo $vesselType | sed -r 's/[:]+/_/g')

                                #strips the path to the last two, includes the ext
                                p1=$(dirname ${pgmarray["$q"]})
                                p2=$(basename ${pgmarray["$q"]})
                                p=$(basename $p1)
                                echo "the value of p1: $p1"
                                echo "the value of p2: $p2"
                                echo "the value of p: $p"

                                #strips the path to the last two, excludes the ext
                                p3="${p2%%.*}"
                                p4=$(basename $p1)/$p3
                                echo "the value of p3: $p3"
                                echo "the value of p4: $p4"

                                #crop function
                                echo "values going into the crop function"
                                echo "$XSIZE'x'$YSIZE'+'$xchg'+'$ychg"
                                echo "$xmin'x'$ymin'~'$xoffset'~'$yoffset"
                                echo
                                convert  -crop \($XSIZE'x'$YSIZE'+'$xchg'+'$ychg\) \
                                        +repage \
                                        -crop \($xmin'x'$ymin'~'$xoffset'~'$yoffset\) \
                                        ~border $optPad \
                                        ${pgmarray["$q"]} "$outDirPath/${p}_${contact}_${modifier}_$p3$ext"
                                #add the original filepaths and the new filepaths to the text file
                                echo "$dirname$p/$p2,$outDirPath/${p}_${contact}_${modifier}_$p3$ext,${p}_${contact}_${modifier}_$p3$ext,$vesselType,$contact"
 >> $outDirPath/'baseDirectory.txt'

                        done #ends inner for
        done #ends outer for
} #ends processFiles function

#####Main Body
dataVal
routeDirname
searchType
processFiles
```

CURE
CONSORTIUM FOR UNDERGRADUATE RESEARCH EXPERIENCE

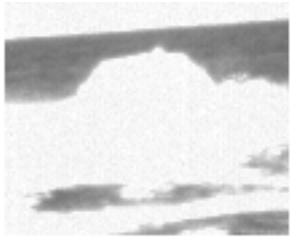
LACC
LOS ANGELES CITY COLLEGE

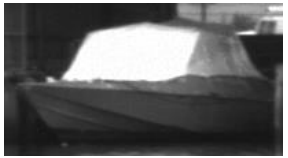
NSF
Work supported by NSF
Grant #1460538

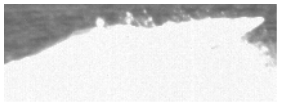
NASA
Jet Propulsion Laboratory
California Institute of Technology

Datasets for Retraining

- Quick Annotator Tool
- Relabeled objects







Work supported by NSF Grant #1460538

Jet Propulsion Laboratory
California Institute of Technology

Acknowledgments

- Professor Paul McCudden for his guidance and attention in the Cure pro gram
- Dr. Michael Wolf for the opportunity at JPL
- Dr. Dan Levine for the conceptual guidance in computer vision and its rel ated fields
- Viet Nguyen for providing technical materials and responding to questio ns
- Dr. Jayesh Bhakta for recognizing the greatness in me

cure
CONSORTIUM FOR UNDERGRADUATE RESEARCH EXPERIENCE

LACC
LOS ANGELES CITY COLLEGE

NSF
Work supported by NSF
Grant #1460538

NASA
Jet Propulsion Laboratory
California Institute of Technology