Criterion B: Design

Tool.cpp

Name	Parameters	Functionality
SplitString	string text, string splitters	It splits the string text at one of the splitters chars

ConfigReader.cpp

Name	Parameters	Functionality
MakeConfigFile		Make a new config file with the date in the program
RemoveConfig		Deletes the config file
ReadConfig		Reads all the important values from the config file
GetConfigVarsFromID	int id	When the config file has been read this function returns the value, given the ID set by the enum.

Trainer.cpp

Name	Parameters	Functionality
Train	String arg	This is called to start the training process
CorrelateImageFeatures	list<feature>* keyFeatures	checks to see if there are any duplicates of different features like a car having two wheels and removes them
GetInputImages	list<string>* imageFiles	Gets n number of training images from user
GetMostImportantPartsOfImage	Mat *grayImage, int maxFeatures, float xStepSizePersentOfImage, float thresholdPresent, int maxfeatureSizeInSteps, int minfeatureSizeInSteps	Essentially just gets the Most Important/Complex Parts Of the image

UseRating	const feature& first, const feature& second	This is used with the build in sort function to get the feature’s “rating” variable.
CorrelateFeaturesCrossImage	list<list<feature>>* featuresForEachImage	Matches keywords from different training images
CreateKeywords	list<list<pair<feature*, Point>>> keywordLists, int thresholdOfSharedImages	For each list in the list it added every image together according to their offset.
SaveMultipleAsImgFiles	string folderPath, string folderName, list<feature>* images	Saves multiple keywords to file
SaveOneAsImgFile	string folderPath, string folderName, feature* image	Saves one keyword to file

OpenCVTools.cpp

Name	**Parameters**	**Functionality**
LoadImages	list<string> paths, ImreadModes im	Loads each image at each path to a list of Mats
ShowImages	list<Mat>* imgs	Shows a list of images using the build in OpenGL imshow method
ShowImage	Mat* img	Shows a single image using the build in OpenGL imshow method
GetGradientImage	Mat grayScaleImage	Uses the sobel operator to make an edge image.
MakeMatFromRange	Point start, Point end, Mat* image, bool ShowImages	Makes a Mat from a portion of another
DoOneCorrelation	Mat *image, Mat *templ, Point p, Mat *result	Gets the correlation of two images at a given offset p regardless of overlap.

DoesCorrelationReachThreshold	Mat image, Mat templ, float threshold, Point *location, bool scaleImg2ToMatchRows	Gets the best correlation finding the offset and checks to see if the correlation passes a threshold.
AddImagesAt	Mat* img1, Mat* img2, Mat* result, Point offSet, Point* mainImageOffSet, float ratio, bool cropEdgesToSquare	Added the images at an offset
TotalMatAddByOne	Mat image	Added every point in the image by 1. This is to make 0 transparent because there is not value for this in a uchar(byte)
CorrelateWithConvolution	Mat src, Mat templ	Uses the matchTemplate method to get the best location
HoughFeatureAccumulater	list<feature> features, Mat src, float PresentOfBinSizeForOutput, int removeRadius, bool ShowImages	Gets all the data associated with places the center or centers could be
UseValue	const pair<float, Point>& first, const pair<float, Point>& second	This is used with the build in sort function to get the pair’s first value.
FindPeaks	Mat image, int numOfPoints, float peakAmountThreshold, int removeRadius	Find the best spots for possible centers.

DrawPointOnImage	Mat src, Point p, Scalar color, int size	Draws a point at point p on the image

Detector.cpp

Name	**Parameters**	**Functionality**
Find	string arg	This is called to start the detection process
GetObjectsToFind		Gets n number of objects to find from user
GetFilePathsFromObjectNames	list<string> objects	Looks at saved keyword data using the names the the keywords were saved under.
CreateFeaturesFromFolderPath	string folderPath	Reads from file to the feature class
DrawOnImage	Mat src, float PresentOfBinSizeForOutput, list<pair<float, Point>> peaks	Draws on the image at the point

Source.cpp

Name	**Parameters**	**Functionality**
main	int argNums, char** argv	Main function, asks you want you want to do with the program train or find

Tools.h feature class

Name	**Parameters**	**Functionality**
GetRating		Rates how complexed the image is using the gradient image

Operator==	const feature& feature1	If something like (feature1 == feature2) is used this checks to see if the grayScale images are equal.

Test plan

To test to see if the program works as intended test images will be used. 3 test images of the same object will be put into the trainer, and then the resulting keywords will be used to find where the object is in the third test image. This plan will test if the program is capable of creating useful “keywords.” If it can store are retrieve these keywords. Also that it can find possible locations of the object. Finally, this plan will test if it can draw on the image to indicate the position of the object.

Words 97