UAV Targeting System

Phil Stavri
UCSD, University of Bristol
pstavri@ucsd.edu ps5657@bris.ac.uk

Abstract

Object recognition is a wide and open problem, in this paper I discuss methods used in solving a subset of these specifically orientated towards target classification as proposed by the AUVSI Student UAV Project. The paper aims at specifically the first stages of computation involving histogram analysis to decide on possible target images and methods and ways of optimizing this stage

1. Introduction

The basis of this project was to find out methods of labeling target images based upon a data set of known targets. By combining several smaller and simpler methods an overall decision of target type, location color and pattern was hoped to be achieved.

This pipeline of requirements obviously needs to be constructed such that the processing of images is only computed on images that have passed previously thresholds and tests, therefore the highest yielding practices should come first. I define highest yielding as the test which offers the largest separation of targets to none targets with the least computational expense.

The basic requirements of features to define for this problem are clear: target presence, shape, location, color, and pattern (alphanumeric character on target). A simple search and general knowledge of current methods in shape recognition and character recognition will tell us that they can be quite computationally expensive and therefore probably not a good base to begin our system with, also they depend on the presence of a target, and thus we should most likely begin with a comparison based method on color. I chose to use a chi squared approach to comparing histogram of known target images with live data using color as the feature for the histogram analysis.

1.1 Related Work

Often the test to see if a proposed method is relevant and theoretically feasible is whether or not there is related or similar work being undertaken. This is of course not the case if the research aims to solve a new or previously unsolved problem. However this is not the case here, this project hopes to look at ways of solving object recognition of known targets using color recognition techniques. Therefore we are looking at a subset of color recognition applied to a very specific field.

A lot of related work in this area can be attributed to Funt and Barnard[1,2] as their work throughout color constancy and the uses of color in recognition has been an extremely useful resource and gives a broad view of the problems and difficulties in dealing with color. In fact they explain, test and trial many of the problems in using color for object recognition, but they are explained in detail later. An area where color recognition and models similar to the ones used in this project is in Kittler's generation of sports cues for annotation [6]. This shows how color recognition can be useful and applied to an object recognition or rather in this case a semantic recognition (something/an event occurring in a scene). Based of the fact that there is current and recent work developing questioning the methods in this area as well as using techniques in an applied sense i feel this approach is very much useful and relevant to what is currently known about color and the opposing methods we could approach this problem from.

2. Color for Recognition

Color for recognizing object presence was decided the best option to start, though the methods on object presence or recognition are vast. For this paper I chose to go with a method of comparisons of histograms of images, this comparator method used the color of the images to decide the presence of a target. The basic concept behind this implementation is that if an image has a similar color build up to the test images then it has the chance of having the target inside it, if there is no similarity it cannot. This however does require a suitably large and varying collection of the target images and will of course allow images that have a similar color appearance through, but it will throw away large portions of images that have very little similarity to the target images.

The difficulty with this approach is finding a color space that offers a good heuristic for the type of application required. To begin with the algorithm was ran purely on RGB images as they were found, this didn't give

the desired results in fact many images became confused quickly and the system was simply unmanageable. The reason for this was that there was a difficulty in viewing the separate color space histograms even for a human, the red channel between two highly varying base images could and would have similar appearance in the individual channels, making it a highly difficult problem for the system to counteract. Once this had been noted an obvious solution was to change the color space. Before trial of color spaces it is clear that a color space which separates the the chrominance and luminance of color would be beneficial for the varying illumination conditions that would be encountered in any real life application. Upon testing this concept with HSV it was found to give a clear distinction between the target and other clutter (i.e background) [Fig 1].

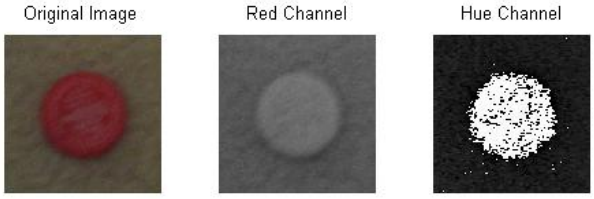

Fig 1 – Color spaces

As the image shows the distinction in the red channel is much more subtle and therefore extremely difficult for the system to detect, hence moving from RGB to HSV gave a substantial boost in quality of the results gained. Yet we should discuss why color does not perform as we may expect, and what exactly is captured by chromaticity and luminance.

Chromaticity is the combination of hue and saturation, being the dominant wavelength in a mixture of waves and relative purity (or amount of white light mixed with the hue) respectively. [5] Luminance is then the brightness of the color perceived. By splitting these attributes of a color we can remove the illumination or the brightness that the luminance refers to in attempt to achieve a color representation that is not overly reliant on color. The HSV representation does help split chromaticity and luminance but of course this does not make it illumination invariant it simply helps that model focus purely at the aspect of color and gives us a less dependent basis on the lighting.

Now we know why this can be useful we should think back to why simply the basic color representation as in RGB would not be useful. This is best described with an example. Imagine a green apple at midday. In the bright almost white light at midday the apple appears green. Now the same apple at say sunset when the sun is low and the light appears almost red still allows us to see that the apple is green. This is due to the fact that color is perceived and very emotive and we can digest this information easily. However a picture during these two different times would show an extremely different representation digitally of the apple. This hopefully should give some intuition as to why RGB as a color space for color recognition is possibly not a good basis. [Fig 1] also shows quite clearly that there is a strong distinction, this distinction is the different in the dominant wavelengths in the image and because it is relative value we see a good distinction, but the red channel is absolute and cannot always give us a good or rather the expected distinction we may have hoped to achieve. Because a color perceptually does not seem to contain red does not mean that in the color representation of RGB that its red channel value is 0. This is counter intuitive and this mixing of palettes causes overlaps and issues where both chromaticity and luminance are represented together, by separating them but merging the color information into hue and saturation we give a better representation for color based recognition.

3. Histogram Operation

The general operation of the system is to randomly select a window from the image, create a histogram from this and compare that histogram to a selection of known target images, if the resulting chi squared values are sufficiently low then we can predict that a target is present.

3.1 Comparative Model

The algorithm uses Chi Squared as the comparative model between two histograms. Chi squared takes two histograms of the same binning and uses a weighted sum of the squared differences between the bin counts of each histogram to give a difference. A low Chi squared result (zero) shows a similarity and a high result (one) shows difference.

The histograms must be normalized to one and in the case that both bins counts sum to zero then an error offset can be added to the chi squared to prevent a divide by zero error. There are other comparative models for histograms other than chi squared, such as correlation and intersection, however chi squared comparison works well in this case as it takes into account the absolute differences between bins and has a useful bias towards bins with low values, so that small differences are not squashed by a large bin count, i.e. A difference of 10 in a

small bin count is weighted more than a difference of 10 in a bin count that is much greater.

Because of this chi squared is a useful and powerful tool for comparing the similarity and differences between two histograms thus is the tool of choice for this problem.

3.2 Windows

The operation of the histograms is extremely important in this project, simply because so many have to be calculated. In the naive case which was first used one histogram was calculated for the entire image and this was compared to a sample target image. However this is clearly a very weak theoretical model, as histograms in nature are based primarily upon quantities and ratio's of items in bins. Take the case where we have an image of a red bottle top on a white carpet with a large border of carpet around the bottle top and we compare this image against a cropped version of the image around the bottle top, the chi squared difference of these images will be substantially different due to the fact that the ratio of bottle top to image size is far greater in the later case. To avoid this issue in the project I have applied a windowing technique to remove this error prone case. The altimeter reading from when the image was taken is used to calculate a window size of the image. This window size is created to be at least 150% of the maximum expected target size and no more than 200%. Working in such a range also helps in the case that very large objects of constant color are discarded as they often throw up very large chi squared results to this window metric.

Upon having windows now I have been able to calculate a histogram based upon the contents of a window. Windowing across the entire image however would be very computationally expensive and most likely unrequired for such a target of a reasonably large size, thus a design feature included has been to allow the windows to overlap by ¼ of their width or height. By doing so we can achieve a complete coverage of the image in searching for a histogram match to the target image, without having the high computation of searching or moving the window by a pixel each time. The issue with windowing is that we have several parameters to choose: window size, window placement and window overlap. As mentioned the size should be due to the altimeter reading based on the physicalitys of image capture, and the overlap has been decided by the idea that if we choose a window that is at least twice as large as the target we would expect to see then by moving that window by ¼ of its width or height we should always keep a significant portion of the target in the window and therefore not misclassify a target because of window splitting. The final problem of window placement comes down to two options. Systematic and procedural, so that we start at the top left an compare and then continue by moving a ¼ of a window right and repeat. The problem here is the target could be anywhere, it is random, yet our procedural model does not take this into account. Hence why i have proposed a random positioning approach to window selection, by randomly choosing windows from anywhere in the image and enforcing the ¼ overlap rule i have found this a much faster technique for finding a target prediction quickly. This leads itself to a greedy algorithm approach, this will be discussed later as it may not be what we want for further computations, but from a purely performance standpoint this idea is feasible and relevant also giving an improvement in execution time over the procedural approach.

Each window then has a histogram created and normalized to 1 which can be used to compare against our known target images, if a sufficiently low chi squared result is found we can stop the algorithm for this image and predict that a target is possibly present, if not we continue until a possible target is found or we achieve the desired density of the search for this image.[Fig 2]

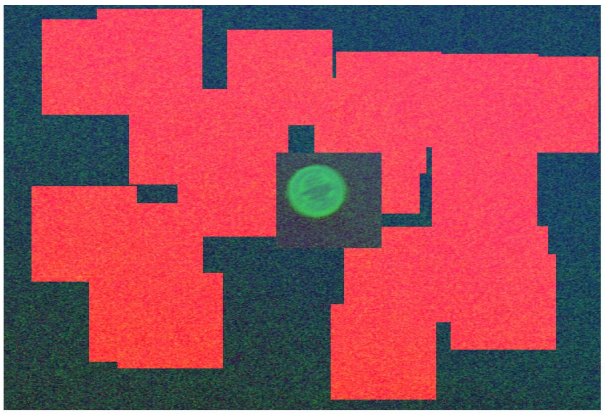

Fig 2 - Windowed Image [Density 25%]

By controlling the window size due to altimeter readings and the density we can control the search of the image, instead of moving the window density in a uniform and standard method i have found a random selection of points to be the most practical for this application as the target can appear anywhere on the image and more often than not a random approach finds the target in a faster time, thus ending the computation on that image in particular and allowing the algorithm to move onto the next image.

3.3 Integral Histograms [3]

An improvement to this system would be the use of integral histograms. This method first dissects an image into a selection of binary images so that there is a binary image for each bin in the histogram. These binary images refer to whether that pixel is in that particular bin or not, thus meaning that the binary images for the bins are all mutually exclusive of one another.

Once these binary bin images have been constructed an integral image[4] is created for each bin. The integral image is the sum of the pixels in the image between that point and the origin (0,0 starting at the top left) in a rectangular box. With these integral images created any further creating of histograms would simply require 4 lookups to an arrays data. Thus making the actual process of creating the histograms at each stage much faster. The binary bins and integral images themselves are quite expensive to create. I found that in my design that the need for integral histograms was not important simply because the time and memory especially in using them outweighed the benefits gained by having this optimization feature for the histograms. However they would become much more useful over when smaller window sizes are required as the constant generating of histograms would add overhead that would give sufficient need for the integral histogram approach to reduce costs throughout processing of the image.

4. Color Constancy

I have found the issue of color constancy to be somewhat of an issue in using color as a means of recognizing targets simply because color can vary greatly in perception to humans in comparison to its engineering value of binary bits in an image. For example in Fig 3, we can see how two different but yet perceptually similar bottle tops in the images shown appear very similar to the human eye yet in the actual color space we lose a lot of information based purely on the constancy of color. This has been difficult to combat and in essence there seems to be a consensus that **"***current machine color constancy algorithms are not good enough for color-based object recognition***"**[2]. However i have found that this is not the case in my examples so far, the results themselves shown an approximate 86% accuracy on the presence of targets with no extra examples of varying lighting conditions, with the extra lighting condition target examples this increases to 95%.

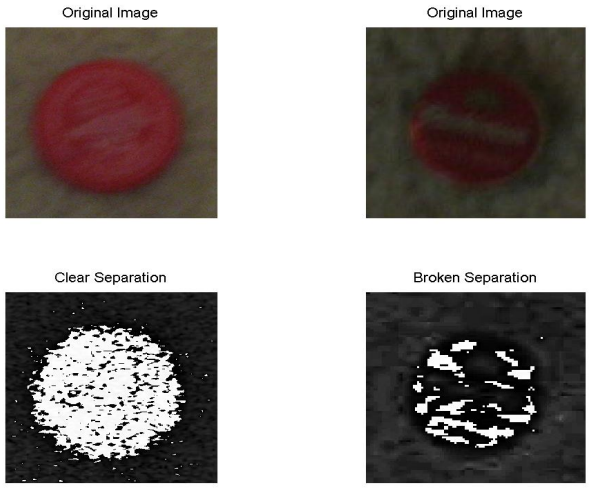

Fig 3 - Color Constancy In Action

The argument follows that color constancy algorithms help in the process of object recognition as they give an improvement in classification results but not nearly enough for an overall effect. Therefore i have chosen to avoid the use of color constancy algorithms such as Grey World[1] and Scale By Max[1] simply because the need in my results seems very minute in the case of color constancy. However there is an issue where several cases where targets are unreported in images due to color constancy issues. This breaks into the trade-off add the complexity of having preprocessed or dynamically processed color constancy algorithms into a system and also the overhead that comes with them to solve in cases 5% of images? For the scale of this project i decided in its infeasibility but i can see the value this would create in trying to ensure a total recollection of the targets in the live data.

5. System Overview

All the stages of the algorithm so far have been explained and contrasted. However it does not give a feel for the purpose of the project in how it all fits together.

The system works based on comparisons to a set of trained images. These data must first be loaded before we can begin. As optimization we should really pre-compute this data otherwise with any substantially large size training set we will simply be incurring added computation that can be saved with little extra work.

With the training images loaded for each of our various illumination differences (4 were chosen, average light, bright light, darkness and a blurred case). These

illumination changes have been used to try and combat some of the effects of color constancy. An ideal world would see the sample training data contain a set of images that ranges over all possible illumination variations and thus perfectly describes our incoming data, however this is clearly not feasible as the selection of data would be far too large to fit into a UAV and also would most likely require a neural network to solve the problem for us. However such a large set is not required, using Chi Squared with some average threshold we can use this smaller subset of training images to return good results.

The threshold are decided in a set and systematic way, first we take a base image that describes the illumination we want to capture for that target. Next we obtain similar images for that illumination. The images must be cropped to a window size manually as [Fig 3] shows. These could be two descriptors of normal or average light. Now we have a number of images for an illumination class we simply compare all the images in this class to our chosen base image, this is simply one of the illumination variation images for this class which we believe best describes the conditions. Chi squared effectively for each of these images is returning us a distance to our base image, by averaging that distance we can get a threshold for allowable images in future.

Yet this is too strict simply leaving it at this perfect average as in this case the average would still have failed some of the training data simply because it is an average, therefore it would be best to loosen the threshold by some chosen amount, as we are not interested if we let through false accepts, but we are positive that false rejects must not happen a higher threshold is not a problem for the system. The thresholds will be applicant and requirement specific but i found in general doubling gives a loose enough threshold to ensure a good bound on the allowed target queries is formed.

Now we have a threshold for each illumination class it is a case of repeating this process but now on incoming query images. An incoming image would be passed of the data stream. It would first be converted into HSV from RGB. Next a random window is selected depending on altimeter readings from the UAV (the higher the plane the smaller the window, due to targets appearing smaller the further away from the UAV they are). Compare the window hue data to the hue data of the base image for each illumination class. It is at this point where we can choose our greediness of the algorithm. If we pass the image as possible containing a target at the first time a chi squared value falls below its respective threshold then we can say th algorithm is greedy, if we continue to search for the best match (helps with location estimation) then it is none greedy this is discussed further in the next section. So we either decide if the image contains a target or not. We when can recurse and keep search more windows (depending on greediness or density of the window we wish to search).

Once the algorithm terminates we then will know whether we found any possible predications of targets, if we did this is the point at which the remainder of target classification would continue, using shape recognition techniques, color detection and letter recognition. If not we can fail the target and begin our target prediction on the next image coming in from the stream.

6. Location Detection

The algorithm as designed above has two possible design choices, greedy or not greedy. I will show that depending on the requirements of the algorithm it can currently increase functionality by locating an estimate of the position of the target it thinks it has located if a match is found.

6.1 Greediness

The greediness of the algorithm matters drastically. And the choice of whether it should be greedy or not comes down to how the algorithm will be used. If we choose the comparison model to act in a way that is greedy what will happen is the first time the algorithm sees a comparison with chi squared that is below the threshold for one of our target types it will predict a target and leave. This is ideal if we wish to have fast performance as once a target is noted we exit happy in the fact that we now know a target is present. But what happens if we want the best match? Based of how the comparison works we want the best match for a target if we are to predict the location. The means we have to ensure that the algorithm does not exit after finding a single target prediction but rather continues until all density comparisons have been calculated.

6.2 Choosing the Location

As this is only an estimate of the location the system only looks at the best match found, and takes the center of its window as the local location of the target. This distance is then calculated from the center of the image. With knowledge about the field of view of the camera, the height and the displacement from the image center by the target we can calculate a GPS location of the target based solely on where the UAV was at the time of image capture. This assumes that the camera is pointing

perpendicular to the earths surface, and due to change of direction this may be out by a little, or by turbulence, yet a gimbal included in the plane should take care of virtually all issues small and minor differences may still take place. Hence why this is only an estimate based of the best prediction of the target and not a greedy prediction, as a greedy prediction could give an inaccurate location that would increase the already likely chance that the estimation will not be perfect due to engineering and physical difficulties. Also to note as the UAV flies higher each increased error in location estimation grows as the high grows, meaning that an inaccurate greedy predication could have hugely differing location results which furthers the need for a none greedy best target prediction to be used for locating the target.

7. Possible Extensions

As often in many project there is not enough time to complete all of what you wished to achieve. This project in particular for me as i have come against several issues i did not expect in the beginning and thus have many extensions i would like to follow at a later stage.

- A PCA based description of targets based upon the sample labeled data, in use of classifying the target shape or type.
- A locater of the target more accurate than simply the center of the target window found, then used in conjunction with the GPS supplied by the AUVSI team's database of the image data to give an accurate GPS location of the actual target location in the world co-ordinates.
- Color descriptors of the target, its base color with the location and the target contours found from previous extension idea's this should be again a classification problem based on the average color within the contours of a shape.
- Letter testing: the AUVSI targets will have alphanumeric characters printed onto them, classification of these is dependent on slant, angle and position, it is in itself a whole new project area and an extension only worth entering i feel if all previous extensions have been completed and honed to a satisfactory degree.

8. Conclusions

In this paper i have studied and explained design decisions, methods and techniques for using histogram comparison to predict the presence or an object in an arbitrary image. From using windowing schemes and the extensions of integral histograms to boost performance and the possible addition or rather conscious absence of color constancy algorithms to give a high percentage performance.

I would conclude overall that color as use a primary detection method for an object is a valid method knowing and understanding certain aspects of the conditions expected to be achieved, constant illumination would be a very pleasant and useful one, however in life this is not the case and thus to use color in such a way a large and viable known target image selection must cover multiple lighting and illumination variances to account for this possibility. Color i have found is a very useful and yet very dangerous tool, assumptions made must be carefully thought through as often many approaches to color can seem useful yet return negative and often random results. In this paper i hope to have explained my experiences of using color as a descriptor to label images with targets and how the problems faced can be overcome by using a variety of different methods available.

9. References

[1] - "Is Color Constancy Good Enough?" - http://www.cs.sfu.ca/~colour/publications/ECCV-98/ECCV-98.pdf -
B Funt et al .
[2] - "Color and Color Constancy in a Translation Model for Object Recognition" - http://www.ece.arizona.edu/~pgsangam/CIC-03-CC.pdf – K Barnard et al.
[3] - "Integral Histogram: A FastWay to Extract Histograms in Cartesian Spaces - http://www.merl.com/papers/docs/TR2005-057.pdf – Fatih Porikli
[4] "Fast Variable Window for Stereo Correspondence using Integral Images" - http://www.csd.uwo.ca/faculty/olga/Papers/cvpr03-a.pdf – Olga Veksler
[5] "Digital Image Processing" 2nd Ed – Gonzalez and Woods
[6] "Generation of semantic cues for sports annotation" - http://www.svcl.ucsd.edu/~nuno/ICIP01/Kittleretal.pdf – Kittler et al.