Low Cost Automatic Object Segmentation by Detecting a Signature Motion Within an Optical Flow Signal

Alan Broun, Matthew Studley
The University of the West of England
Frenchay Campus
Bristol, BS32 0BW
{Alan.Broun, Matthew2.Studley}@uwe.ac.uk

Abstract—Induced motion provides a cheap and reliable way of segmenting an object from the background within an image. As such it has potential applications to a wide variety of machine vision systems. This paper presents a system that segments an object from the background by inducing a sinusoidal motion and then searching for the signal in the resulting optical flow. The effectiveness of the system is evaluated using ROC curves and shown to be reliable and robust in the presence of distractor motions.

I. INTRODUCTION

OBJECT segmentation is a key problem within machine vision, and a variety of approaches are taken to make it more tractable for machine vision systems. Key techniques involve structuring the environment and lighting, and fixing camera positions as much as possible in order to reduce the complexity of the segmentation problem [1].

In situations where the machine vision system is able to induce motion in the object to be segmented we have access to another method to tackle the problem. A known motion can be induced in the object and then evidence of this motion can be looked for in images containing the object. Portions of the image with optical flow signals which closely match the input motion signal can be said to have a high probability of being part of the object of interest.

The motivating example we look at in this paper is case of a robot wishing to visually identify its gripper so that it can build a model of the gripper in order to aid in future manipulation tasks. We believe that the method is general enough however that it could be applied in a cost effective manner to a wide variety of machine vision systems.

In the case of a robot identifying its gripper, it could be argued that for a well engineered robot, for a given joint configuration the position of the gripper in the camera frame can be obtained via calculation. Forward kinematics can be used to find the position of the gripper and camera in Cartesian world space and then a perspective transformation can be used to determine the appearance of the gripper on the image plane. However, such accuracy comes at a price, both in terms of the hardware cost (which must be able to be driven accurately and repeatably to the set positions) and the engineering effort which must be expended to setup and calibrate the system to the required level of accuracy. Using motion to detect the gripper offers a cheap and flexible method that imposes far less rigorous constraints on the required accuracy of the system.

The rest of the paper is organised as follows; Section II outlines previous work in this area, Section III describes the method of detecting the robot’s gripper in some detail and shows how the performance of the system can be quantitatively assessed by generating Receiver Operator Characteristic (ROC) curves. Section IV presents the results of experiments that were carried out to investigate the performance of the system and finishes with a demonstration of how the data gathered by the robot can be used to build a model of the gripper to aid in future detection attempts. Finally, Section V presents concluding remarks.

II. PREVIOUS WORK

The method of using motion, and in particular optical flow to locate a robot’s gripper was presented in [2] as part of work on the robot COG at MIT. A similar technique was also presented in [3] although in this case the assumption was made that the gripper was the only moving object in the scene and thus the technique was unable to cope with distractor motion. Neither of the previous papers presented the technique in much detail as it wasn’t the main research that was being reported.

In this paper we extend the work of [2] by examining the technique in more depth and exploring in detail how changing aspects of the signal or detection process can affect the effectiveness with which the gripper is detected. We demonstrate originality by making use of ROC curves to examine the performance of the system so that different ways of detecting the signal can be quantitatively compared.

The generation and comparison of ROC curves is a technique that is used widely in signal detection theory [4]. ROC curves have also been used widely in the medical literature as a way of evaluating diagnostic tests and more recently Fawcett and others have popularized their use in the Machine Learning literature [5].

III. METHODOLOGY

The work presented here was carried out using a Lynxmotion robot arm coupled with a web-cam delivering images at a
resolution of 320x240 and at a rate of 30 frames per second. This equipment is cheap and readily available demonstrating that this technique can be used to produce good results without having to rely on expensive and highly accurate components.

The robot was programmed to try and detect its gripper in the following way. A detection episode is started with a delay where the gripper is held still. The robot then waves its gripper back and forth by passing a repeating sine wave through its wrist joint. There is then another short delay before the robot searches for signs of the input signal in the optical flow that was gathered during the detection episode. A sine wave was chosen in preference to some other form of motion as it was felt that it would stand a good chance of being recognisable following the transformation from joint space to image space.

For calculating the optical flow between 2 images we use a block matching algorithm provided by the OpenCV library [6]. This allows us to estimate the optical flow for an image by defining a uniform grid of blocks over an image (8x8 pixels for example). The algorithm then provides an estimate for the direction in which they moved in the following frame by searching around the centre of the block for the best match in a SAD (Sum of Absolute Differences) sense. To reduce the amount of computation required, the maximum distance for a block match is limited to 8 pixels for both the x and y direction.

Figure 1 shows an example of the optical flow returned by OpenCV and Figure 2 shows an input sine wave along with the output signal from a block that contains part of the gripper.

In order to determine if a block is part of the gripper, the optical flow signal for a block over the whole of the detection episode is cross correlated against the input signal. This involves treating the signals as random processes and repeatedly calculating the correlation coefficient of the input signal and delayed versions of the optical flow signal. The correlation coefficient is given in [7] as

\[ \rho(X, Y) = \frac{cov(X, Y)}{\sqrt{var(X)var(Y)}} \]  

(1)

Where \( cov(X, Y) \) is the covariance of the 2 signals and \( var(X) \) is the variance of a signal. Essentially equation 1 multiplies each corresponding sample of the two signals together and adds them up, if the signals match then you end up with a large positive number (matching troughs also add to this as a negative times a negative gives positive) and if they don’t match then the peaks and troughs tend to cancel out to give a value close to zero. If the signals are the inverse of each other (peaks match troughs) then this gives a large negative number. By using the covariance of the 2 sequences and dividing through by the square root of the variances we guarantee the the correlation coefficient is in the range -1 to 1. The proof of this is given in [7].

An optical flow block is declared to be part of the gripper if there is a peak in its absolute correlation coefficient greater than some threshold. Calculating the cross correlation of all the optical flow signals for the entire duration of the input signal is an expensive operation so we want to reduce the number of cross correlation coefficients calculated if possible. Fortunately we know that the time delay should remain fairly constant for any robotic system (although we don’t know what the time delay is). Therefore, in our implementation we chose to only delay the input signal by a maximum of 1 second when calculating the correlation coefficients and performance could probably be improved by tightening the window still further.

Figure 2 also shows the cross correlation calculated for the optical flow block along with the peak that we’re looking for.

A. Building a ROC Curve

To allow the performance of the gripper detector to be evaluated, a ground truth labelling of which blocks were part of the gripper and which blocks were not was created by hand. Due to the dimensions of the optical flow blocks, some blocks are only part filled by the gripper so a subjective decision had to be made about about what the ground truth gripper segmentation should be. The ground truth segmentation used for all tests in this paper is shown in Figure 3.

For each test sequence the ROC curve was calculated using an efficient method given in [5]. We then use the ROC curve to evaluate the classifier by looking at the Area Under Curve (AUC) statistic.

As its name suggests the AUC is the area under the ROC curve. As a ROC curve occupies the unit square the AUC takes
values in the range \([0,1]\) and in general the higher the AUC the better the classifier.

A classifier cannot however, be evaluated reliably from a single ROC curve as the ROC curve is generated from just one set of test data. Therefore whenever we perform a test we create 5 versions of the same test sequence and average the ROC curves together using the vertical averaging technique given by [5].

B. Choosing a Threshold

ROC curves are good for evaluating classifiers as a whole but when a decision has to be made a threshold or operating point has to be set. The choice of which operating point to choose for a classifier varies based on the cost of making an incorrect classification. This may not be equal for the 2 possible classifications. The classic example is that for a test designed to detect the presence of cancer, diagnosing a healthy person with cancer might be a lot less costly (in human terms) than failing to diagnose a person who did have cancer.

In our case we assume that making an incorrect classification has the same cost for both a false positive and a false negative and therefore seek to maximise accuracy which is given by [5] as

\[
\frac{TP + TN}{P + N}
\]

Where \(TP\) (true positives) is the number of positive (part of the gripper) blocks that have been classified correctly, \(TN\) (true negatives) are the number of negative blocks that have been classified correctly, and \(P + N\) is the total number of blocks in an image, the sum of both positive and negative blocks.

C. Experiments

A variety of tests and experiments were carried out to explore how well this method of locating the gripper worked and to determine how robust it was. We first compared the optical flow blocks identified as being part of the gripper with a ground truth representation, and used this to tune the behaviour of the classifier. After that, we looked at whether smoothing could be used to compensate for some of the inevitable noise in the optical flow data and improve detection. We then explored how robust the technique was in the presence of distractor motion, both real world and artificially added. Finally, we used the data gathered from the motion detection phase to build a simple Histogram based model of the gripper. This last item was an attempt to show the technique could be used as a tool by the robot in order to gather information for a higher level model building process.

When carrying out the experiments, the camera was kept fixed, and the robot’s gripper was moved to a pre-programmed position before it was waved in order to detect it. Where possible the experiments were carried out using recorded data. This allowed experiments to be repeated easily and with small parameter changes whilst guaranteeing that everything else remained the same. When recording data for experiments we recorded both the output from the camera and the commanded joint angle that was sent to the robot’s wrist. The gripper movements were padded with 2 seconds of non movement both before and after the gripper wave.

IV. Results

A. Detector Performance

In general the detector worked well and was able to identify a large number of optical flow blocks as being part of the gripper in agreement with the ground truth, whilst rejecting blocks which contained noise or motion from a source other than the gripper.

As an example we look at the simple case of the robot waving its gripper 3 times in order to detect it. In this particular situation the gripper is the only thing moving in the sequence i.e. there are no distractors. The averaged ROC and accuracy curves are given in Figures 4 and 5 respectively.

![Figure 4. ROC curve with 95% confidence intervals obtained by varying the classifier threshold](image)

Choosing the maximum accuracy we get a threshold of 0.57 and Figure 6 shows the result of applying that threshold. We can see that the majority of the blocks given in the ‘ground truth’ were detected as being part of the gripper and the main errors come either from areas with low texture (where the movement isn’t detected) or else from areas that the gripper moved into whilst it was waving.
0.0
0.2
0.4
0.6
0.8
1.0
Accuracy
0.0
0.2
0.4
0.6
0.8
1.0
Threshold

Figure 5. Accuracy graph for the classifier with 95% confidence intervals. The threshold at which the maximum accuracy is obtained is marked with a vertical line.

Figure 6. Optical flow blocks matched to the input signal by the classifier with a threshold of 0.57

Figure 7. Plot of AUC varying with the number of waves

B. Varying the Input Signal

It seems reasonable to suppose that the longer the robot the robot waves its gripper for, the more likely it is to be able to distinguish it from the background. To test this theory we recorded one long sequence with 9 waves of the gripper and progressively cut out the later waves to produce sequences with different numbers of waves. This guaranteed that a classifier using 9 waves would have exactly the same information as a classifier using fewer waves, plus the extra information for the additional waves.

The AUC of each of the classifiers is compared in Figure 7. It can be seen that as expected, increasing the number of waves has a small positive effect on the AUC of the classifier. This benefit gets progressively smaller as more waves are added.

C. Smoothing the Input Signal

One thing that becomes immediately obvious when looking at the raw optical flow signal for a block (see Figure 2) is that there is a lot of noise in the signal. This could be for a number of reasons, i.e. the block based optical flow algorithm gives only a coarse sampling of the actual optical flow field and the algorithm can be fooled by parts of the image which look similar to other nearby parts of the image. Also, in our case the trajectory for the gripper is implemented by sending out a series of set points to the servo motors and it appears that the camera captures points where the gripper is stationary causing the optical flow to drop to zero.

Whatever the cause of the noise, it obscures the signal we’re trying to detect. In an attempt to compensate for the noise we filtered the 1D optical flow signals with varying levels of Gaussian smoothing. Figure 8 shows the result of applying Gaussian smoothing with varying values of σ to the optical flow signals for the variable number of waves test data. This seems to indicate that a small amount of smoothing does improve the detectors performance but too much will start to degrade it.

D. Robustness in the Presence of Distractor Motion

One advantage of this method for identifying the gripper that was reported by [8] is its robustness in the presence of other distracting motion which may be occurring whilst the robot tries to locate its gripper. To investigate this we carried out tests with both real and artificial distractor motion.

The real distractor motion consisted of the author and others moving round and producing random motion in the background. For the artificial distractor motion, moving blobs
were added to the test data after it had been recorded. The blobs were randomly textured to make sure that they were accurately detected by the optical flow algorithm, and they were moved with random periodic motions.

For each type of distractor (none, artificial and natural) 5 tests were run and the average AUC ± standard deviation is shown in Figure 10. The results of the tests with distractors were compared to the tests with no distractors using Student’s t-test for unpaired observations, and the p value for these comparisons is also shown in Figure 10. This shows that although the mean has gone down for the tests run with distractors, the high values of p (> 0.05) means that it is not a statistically significant drop. Also, the high AUC values for the tests done with distractors support the assertion that the detector is capable of reliably segmenting the gripper from the background even in the presence of distractors.

Figure 9. Artificial distractors

Figure 10. The mean and standard deviation of the AUC for varying levels of distractors. Also shown is the result of applying Student’s t-test to compare the samples

E. Building a Model of the Gripper

The work presented so far in this paper has shown that we are able to identify the gripper fairly accurately by looking for evidence of a signature movement. It would be fairly cumbersome however, if the robot was required to go through the extended detection process every time it wanted to determine the current position of its gripper in its field of view.

Therefore, the next stage is to use the data that we’ve gathered from the detection process - the parts of the screen which the robot believes with high certainty contain the gripper - to construct a model of the gripper that can be used to identify it using far less effort.

The literature contains a large number of methods that we could use to construct a model of the gripper. Examples include

- histogram methods which estimate probability distributions for characteristics of the object i.e. colour[9] or texture [10].
- feature based methods which model an object as a collection of distinctive features that result from applying a feature detector such as the SIFT operator [11].
- methods which store the boundary/silhouette of the object in a compact form i.e.[12]

Once a model has been constructed of the gripper, this model can be used by the robot to search for the presence of the gripper in its field of view without it having to wave the gripper around.

To prove that a useful model of the gripper can be constructed we choose to build a histogram of the gripper in RGB colour space to act as a model for the gripper and then use the CAMShift algorithm of [13] to track the gripper around the screen.

All of the motion blocks which were declared to be part of the gripper by the gripper detection process were assumed to contain only pixels from the gripper. The contents of these blocks were used to build an RGB histogram with 32 bins in each dimension. The histogram was used as the probability distribution for the colour of pixels in the gripper and this was back projected onto image frames to give a probability value for each pixel that estimated how likely it was to be part of the gripper. The CAMShift implementation from OpenCV is then used to find the local maximum of this distribution and thus track the gripper. Figure 11 shows that the CAMShift was able to track the gripper accurately although obviously it may have had problems if there was another large object in the scene which was the same colour as the gripper. We are not trying to suggest that the histogram based model of the gripper is the best that could be created however. Instead we are demonstrating that the method of segmenting an object from the background using motion provides ample information for a good model to be constructed.

V. CONCLUSIONS

We have presented a method for segmenting an object from its background by inducing a known motion in the object. The method has been used to enable a robot to identify and build a model of its gripper, and this implementation has been evaluated and found to be reliable and robust to the presence of distractor motions.

REFERENCES

Figure 11. Gripper being tracked using the CAMShift algorithm. The grey level value of each pixel indicates its probability density in the histogram and the orange rectangle is the track returned by the CAMShift algorithm.