The final publication is available at link.springer.com

Combining depth and intensity images to produce enhanced object detection for use in a robotic colony

Steven Balding[1] and Darryl N. Davis

Faculty of engineering and computer science, University of Hull.
S.Balding@2013.hull.ac.uk
D.N.Davis@hull.ac.uk

Keywords: Anchoring, Robotic Vision, Sobel, Depth map, Robotic Colony

Abstract. Robotic colonies that can communicate with each other and interact with their ambient environments can be utilized for a wide range of research and industrial applications. However amongst the problems that these colonies face is that of the isolating objects within an environment. Robotic colonies that can isolate objects within the environment can not only map that environment in detail, but interact with that ambient space. Many object recognition techniques exist, however these are often complex and computationally expensive, leading to overly complex implementations. In this paper a simple model is proposed to isolate objects, these can then be recognize and tagged. The model will be using 2D and 3D perspectives of the perceptual data to produce a probability map of the outline of an object, therefore addressing the defects that exist with 2D and 3D image techniques. Some of the defects that will be addressed are; low level illumination and objects at similar depths. These issues may not be completely solved, however, the model provided will provide results confident enough for use in a robotic colony.

1 Introduction

Robotic colonies are an important part of modern robotics research and are becoming used more and more in industry. A robotic colony provides a collection of agents that can work uniformly and autonomously to survey and interact with their environment [1]. However for a robot to interact with other agents and the ambient environment the robot must find a relationship between the perceptual data it gathers and a symbolic representation of objects within the environment [2][3][4]. The perceptual data gathered is only useful for anchoring, however, if an object can be isolated within the data. Methods proposed for isolating object within an image are often computational expensive and complex, with techniques such as Convolution Neural Networks becoming popular[5]. However for a robotic agent to detect objects within the environment simpler methods, such as kernel filters, can be used to separate and classify objects by features or light intensity. These less complex methods have issues which are addressed in the paper.

Detecting objects within a 2D image is a common practice with good results [6] [7] achieved using Kernel filters such as Sobel operators providing not just edge detection but also the orientation of the edge [7][8]. Vision processing using just Sobel filters can handle image noise relatively well, and as such, have been commonly used for quick moving detection, such as license plate recognition [9]. Canny edge detection is another popular method that gives a more refined edge information at the expense of computational time [10][11]. This technique has been used in conjunction with convex hulls to remove isolate objects form the background [12].More expansive techniques allow an object to be described in 3D, using binocular images to estimate distance. However both 2D and 3D techniques suffer from similar problems. The images need to have a contrast on the edge for an edge to be found. The images being analysed might not yield edges due to images with little to no definition or object occlusion and even reducing noise with a Gaussian distribution filter could leave the image with edges that are not identified. There have been 2D and laser range finder combinations [1] used to success, however this will still leave problems with object occlusion and does not account for the 3D world.

In contrast 3D cameras, such as the Kinect, are being used commonly for computer vision [13]. The resulting image form a 3D camera can still be a 2D matrix but rather than an intensity or a colour, instead there is the depth of any given x, y pixel. Plotting this result as a 3D point cloud, objects can be identified and mapped using the resulting vectors. However, problems still arise when considering an image that is contains multiple objects at the same depth, classifying these with depth alone is a rather difficult task that requires high resolution equipment.

In this paper, a model is proposed incorporating simple 2D and 3D techniques for producing a probability mapping of an objects outline, this data can then be used to isolate an object and tag that isolated object. The model will achieve this by normalize the colour based image and the depth image and run separate filters on the receptive image to detect edges within each image, the intensity of the edge will also act as our probability of an object at that point. Analysing the results of both image filters will produce a probability map that will show where an object or its boundary is likely to be regardless of stacked objects or poor lighting, while being computationally inexpensive and simple to implement. The object can then be classified based on simple pre-defined characteristics such as volume or color.

2 Sobel Operator

2.1 Overview of Sobel operator

When detecting edges in an image, often the process is approximating the first order derivatives by convolving a set kernel with the original image. With a Sobel operator (or filter) [14], kernels are used for both the horizontal and vertical plane, as shown in Figure 1. This will return the local derivative of that kernel for vertical and horizontal respectively. To obtain the absolute magnitude edge gradient of that pixel:

$$G = \sqrt{G_x^2 + G_y^2}$$

Gx = Left Sobel filter.

Gy = Right Sobel filter.

Because the horizontal and vertical plane are observed it is possible to also find the orientation of that edge pixel with:

$$\theta = \tan^{-1} \frac{G_y}{G_x}$$

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Table 1. Kernel for Vertical (left) and Kernel for Horizontal (right)

After these operations, there will be a maximum response from the vertical and horizontal plane, a final edge gradient and angle of orientation. This spatial information is very useful for identifying an object. Not only do these operations leave us with a clear edge, the angle of neighboring pixels can be observed to find recognizable shapes and curves; this will further help isolate object in a picture. This operation also includes average factoring which helps reduce some of the low level noise of the image by taking a more balanced view of the entire image.

For any modern robot this will also not prove to be too computationally expensive as the kernels can be decomposed as products of an interpolation and a differentiation kernel, and could even be possible from a video feed.

There are issues however, and these issues fall into two main categories; the first being image noise and the second being intensity of edge being detected.

2.2 Problems of Sobel operators

The problem of noise in images is a common one - image noise can come in many forms, usually coming down to white noise or salt and pepper noise. Sobel operators, as stated before, can naturally deal with some amount of noise and post-processing with filters such as Gaussian which can further reduce the image noise and better detect edges. Methods for Sobel noise reductions have been proposed [15], but these can be computationally expensive with a lot of post or pre-processing. In the model proposed, noise will be reduced by comparing edges from 2D and 3D images.

Edge detection, at its core, is looking at the difference of intensity of neighboring pixels in order to find an edge. For this to be successful, a clear change in light intensity is required. Considering an image that is only partial lighted, will show sections of the object within the image that produce a false edge.

There are ways to deal with this in the context of robotic agents identifying the object, for example using neural networks or convolution networks might mean training an image to be recognised from a partial image. However this could take a lot of sample images and long training time, more over this could still be inaccurate if there is a large percentage of the image covered in darkness or the shape left behind could look like another object.

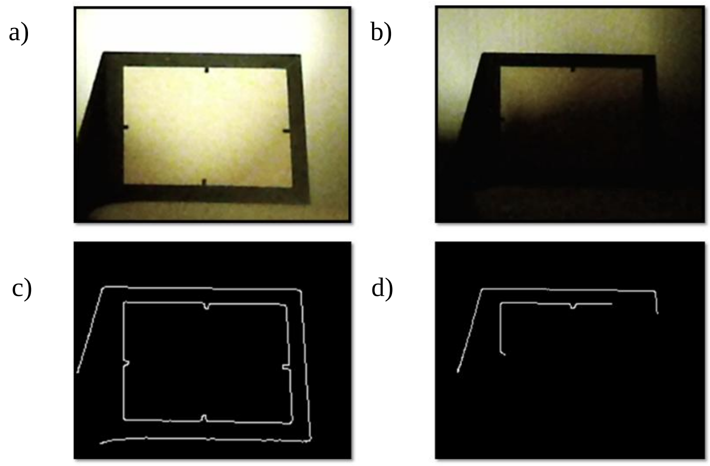

Fig. 1. a) Good lighting b) Poor lighting c) Good edge detection d) Poor edge detection

It should also be noted that it is not just light that can cause edge detection to be unreliable, stacked objects of the same or similar colour can sometimes cause issues.

The system proposed will not intrinsically solve this problem, although dues to its probabilistic nature, it could highlight areas of potential for a more thorough inspection. It should be considered, however, that two object at the same distance same colour and out of reach of haptic feedback would be difficult for us as humans to differentiate between, so expecting a robot to perform that type of recognition would be very difficult.

3 Depth Image mapping

3.1 Depth Image Overview

3D depth cameras offers robotics sight with reasonably accurate depth analysis. This one of a few major branches in robotics and AI that does not consider how a human perceives the world, but rather uses technology to surpass that of our vision. It is still a relatively new field for computer vision, with most offers in the past favoring 2D images, but with the reduction of price of the cameras, more and more example of vision processing with RBG-D images are occurring.

3D cameras will return a depth value fort each x and y in an image, this is usually converted to either an intensity value, giving a grayscale image of depth or a point cloud, where by the pixels are mapped in 3D Cartesian space.

In a paper regarding depth kernels [16],one of the feature detection methods mentioned is edge detection over depth maps, and although the results are good, the problem of object occlusion and same depth objects is not covered.

There are other examples of 3D object classification that use point could data to good effect. Some opting for measuring similarities between point descriptors of curated 3D models and real world point cloud data [17] and some using custom 3D Convolution Neural Networks (CNN) [5]. These approaches are very successful, however they are computationally complex, and, in the case of the CNN, they lack information contained within the pixel.

It is proposed that the model in this paper will help reduce computational complexity, store angular and spatial information of a given pixel and produce a strong probabilistic assessment of an objects border. It is also still unclear in these results if this type of network or feature descriptors can separate and classify stacked and same depth objects

a) c)

b) d)

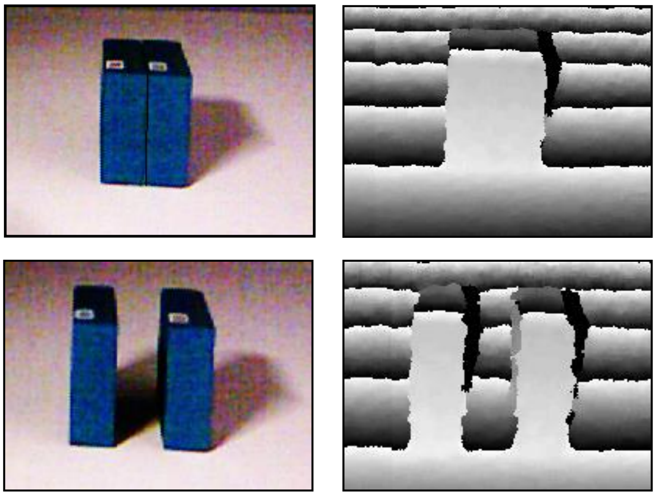

Fig. 2. a) Colour picture of two stacked objects **b)** Depth map of two stacked objects **c)** Colour picture of two separated objects **d)** Depth map of two separated objects

3.2 Sobel filter with depth map

In this paper the model that is proposed will take edge detection values from a 2D image using a Sobel filter, but also values from an RGB-D image of the same perspective of the original 2D image. To achieve this there are 3 stages;

First is to downsize the image from the camera to the same resolution of the RBG-D images. In the case of a Kinect V2 camera (V1 performs poorly in low light conditions) this is 512 x 424. The colour image can be recorded at 1920 x 1080. The Kinect has a built in function to deal with this conversion called the 'ICoordinateMapper', other cameras might have to manually preform registration.

Then a grey scale images will be produced where by the intensity of light (from 0 – 255) of any given pixel will be the value of the depth of that pixel. The Kinect gives precision of 11 bits ($2^{11} = 2048$), this means that the pixel depth can be scaled to the light intensity value which is 0 – 255. When this has been processed, the resulting image will be a depth map similar to that of a grey sale image, but rather than considering light, depth is the focus.

Fig. 3. Image of grayscale depth map

Finally, running the Sobel operator over the created depth map, treating it as before. This will yield a value of an edge that is not based on the derivative of light intensity (although we are creating an artificial grey scale for the Sobel filter), but rather a response to the difference on the depth of an image. The values given back from this operation that highlight edges on object that are not illuminated, something that cannot be ascertained from just processing a 2D image.

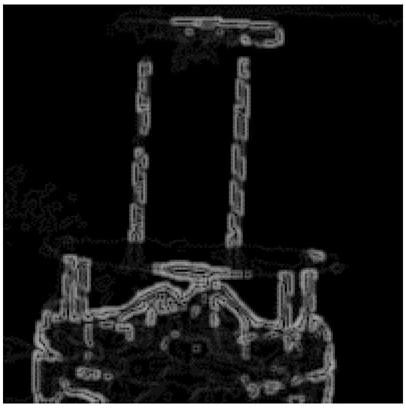

Fig. 4. Image of filter over depth map

After obtaining an edge detection image on both a 2D light intensity image and 3D depth map, the data points for both images can be combined to ascertain the probability space.

4 Probability Mapping

4.1 Probability weighting

At the point of obtaining the two images of the same resolution, it is possible to take the newly found edge intensity value and combine both value to form the likelihood of that edge existing. However, the higher the intensity the more weight should be placed on its value, moreover, if the light source for the image is good then weighting more towards these probabilities would act as more realistic impression of an objects edge, as using the difference in light intensity is more reliable than just depth detection.

Object in a real dynamic environment tend to collide and often exists at very similar depths (for example, a shelf of books). A rudimentary approach is proposed to account for the quality of the light source and weight the values in favour of well illuminated images.

To determine the weighting of the probability map the over quality of the light source must be determined. In this paper, the proposed method is basic, more complex methods could be implemented for better accuracy. To obtain the light source rating a relative luminance calculation will be performed on each pixel in the original colour 2D image, these values will then find the arithmetic mean to find a finale weighting for the probability mapping.

$$L = \frac{\sum_{i=1}^{q}(\sum_{j=1}^{q} 0.2126R_{ij} + 0.0722B_{ij} + 0.7152G_{ij})}{q}$$

q = Pixel count of the image

R = Red pixel intensity for given x,y coordinate
B = Blue pixel intensity for given x,y coordinate
G = Green pixel intensity for given x,y coordinate

After the calculating the coefficient for the weighting, it is possible to produce the probability map using the following to give a scaled balanced return.

$$\frac{a * \left(\frac{l}{255}\right)^2 + b * \left(\frac{d}{255}\right)^2}{a + b}$$

a = weighting for 2D image
b = weighting for 3D image
l = pixel intensity for 2D (light based) image.
d = pixel intensity for 3D (depth based) image.

The weighting on the combined probability calculation will be set for 5 for depth map images and scaled from 1 – 10 for the relative luminance, this will account for an image with high and low quality light sources. The depth could be called higher, maybe as a result of objects distance from the camera.

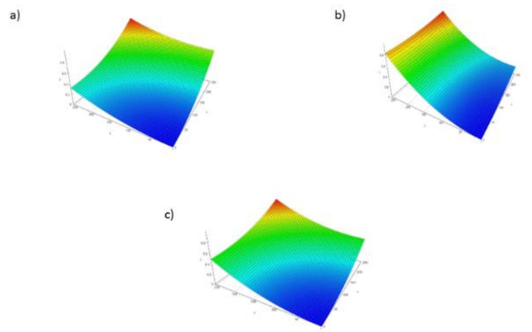

Fig. 5. a) Graph of probability map a = 5, b = 5 b) Graph of probability map a = 10, b = 5 c) Graph of probability map a = 1, b = 5

4.2 Object detection

There are many different methods for recognizing an object from edge detection [18] and more complex methods that will deal with partial and complex outlines [19], the output from the model proposed in this paper could also fit into other models, using the probability map instead of an edge intensity map. However, because the spatial and orientation data is kept (due to the Sobel operator) for each pixel in both the light and depth map image, a 5 dimensional set can be produced for any given pixel in the image:

$S = \{p, l, l\theta, d, d\theta\}$

S = Set of values stored within each pixel
p = Value from probability map at given X, Y.
l = Value from 2D Sobel filter map at given X, Y.
lθ = Orientation angle from 2D Sobel filter at given X, Y.
d = Value from 3D Sobel filter map at given X, Y.
dθ = Orientation angle from 3D Sobel filter at given X, Y.

With this complete information about any given pixel it is possible to classify an object in an image more clearly. An example; a football is taken as the object, the ball is white and written in the center, in black, is the name of the manufacturer.

Given only 2D edge detection producing a convex hull of this image might produce one for the outline of the ball and another for the writing, with the information given in this model it is possible to ask, is this writing the same depth as the object it is placed on. This extra level of information can further help isolate, classify and tag an object in an image and further more reduce high level noise.

5 Results

The probability map that is returned from the depth and colour images can be placed through a threshold operator, in this case hysteresis, to return an images that gives a solid line that represents the object. An advantage of this probability map is that the thresholds can be adjusted to suit the needs, using the probability of a line becoming the threshold. This final image can then be classified from features such as convex hull area, corners, histogram features or many other feature descriptors, in this case we use a simple convex hull area technique.

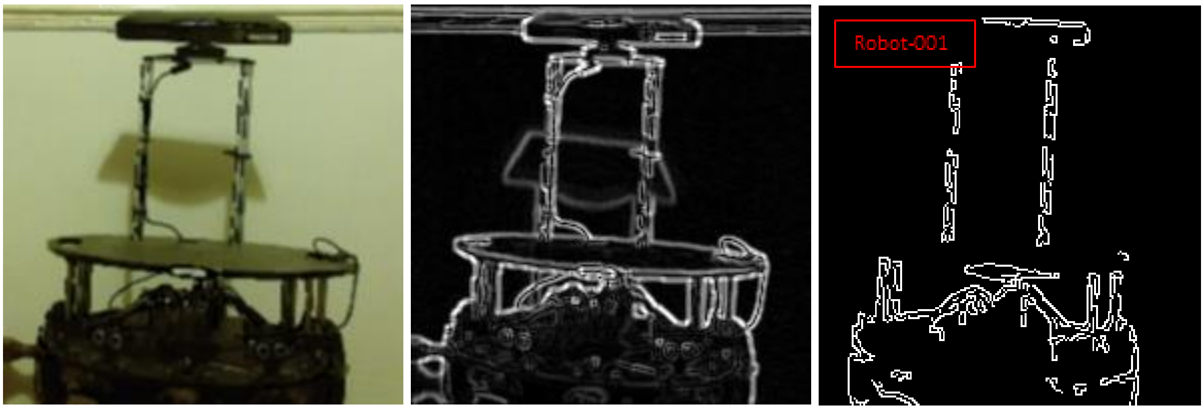

Fig. 6. Left: colour image. **Middle**: Probability mapping. **Right**: Finally tagged image.

Another aspect of this model was it simplicity, not only to implement, but for computational speed. Below are the results of a timed run using a simple Sobel operator as the benchmark.

	Test 1 (Microseconds)	Test2 (Microseconds)	Average (Microseconds)
Benchmark (just Sobel)	359,329	341,335	350,332
Depth map	296,546	278,512	287,529
Probability mapping	362596	382596	372,596
Fully Classified	420,923	413,519	417,221

Table 2. Showing task complete times, Measured with the Chrono library

6 Conclusion

The model presented in this paper should find use as a simple and quick to implement solution for multi agent swarm. This model can classify images quickly and loosely, whilst making up for some of the issues of the incorporated techniques.

One of the advantages that this system conveys is, each pixel can still contain depth, light intensity and orientation within each pixel. This in turn meant that classification algorithm can work with more than just the outline of an image.

In further work objects will not just be classified as simple objects, but rather, account for the potential of sentience in the perceived data. This would add another dimension of functionality, as well as adding further accuracy to the classification.

7 References

1. Milella, A., Di Paola, D., Mazzeo, P.L., Spagnolo, P., Leo, M., Cicirelli, G., D'Orazio, T.: Active Surveillance of Dynamic Environments using a Multi-Agent System. IFAC Proc. Vol. 43, 13–18 (2010).
2. Harnad, S.: The Symbol Grounding Problem. Symb. Grounding Probl. D. 42, 335 (1990).
3. Coradeschi, S., Saffiotti, A.: An introduction to the anchoring problem. In: Robotics and Autonomous Systems. pp. 85–96 (2003).
4. Gwatkin, J.: Anchoring in a Cognitive Robot. Mach. Learn. (2009).
5. Maturana, D., Scherer, S.: VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. Iros. 922–928 (2015).
6. Sharifzadeh, S., Centre, E., Manufacturing, I.: Edge detection techniques : Evaluations and comparisons Edge Detection Techniques : Evaluations and Comparisons. 2, 1507–

1520 (2008).
7. Kim, D.: Sobel Operator and Canny Edge Detector ECE 480 Fall 2013 Team 4. 1–10 (2013).
8. Anusha, G.: Implementation of SOBEL Edge Detection on FPGA. Int. J. Comput. Trends Technol. 3, 472–475 (2012).
9. Israni, S.: Edge Detection of License Plate. 3561–3563 (2016).
10. Lakshmi, S., Sankaranarayanan, D.V.: A study of Edge Detection Techniques for Segmentation Computing Approaches. Int. J. Comput. Appl. CASCT, 35–41 (2010).
11. Pavithra, C., Kavitha, M., Kannan, E.: An efficient edge detection algorithm for 2D-3D conversion. In: 2014 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC). pp. 434–436. IEEE (2014).
12. Adhikari, S., Kar, J., Dastidar, J.G.: An automatic and efficient foreground object extraction scheme. Int. J. Sci. Adv. Inf. Technol. 3, 40–43 (2015).
13. R.Abdulmajeed, W., Zuhair Mansoor, R.: Implementing Kinect Sensor for Building 3D Maps of Indoor Environments. Int. J. Comput. Appl. 86, 18–22 (2014).
14. Kanopoulos, N., Vasanthavada, N., Baker, R.L.: Design of an Image Edge Detection Filter Using the Sobel Operator. IEEE J. Solid-State Circuits. 23, 358–367 (1988).
15. Gao, W., Yang, L., Zhang, X., Liu, H.: An improved Sobel edge detection. Proc. - 2010 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. ICCSIT 2010. 5, 67–71 (2010).
16. Liefeng Bo, Xiaofeng Ren, Fox, D.: Depth kernel descriptors for object recognition. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 821–826. IEEE (2011).
17. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing Objects in Range Data Using Regional Point Descriptors. Comput. Vis. - ECCV 2004 8th Eur. Conf. Comput. Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part III. 3023, 224–237 (2004).
18. Ushma, A., Scholar, M., Shanavas, P.A.R.M.: Object Detection In Image Processing Using Edge Detection Techniques. IOSR J. Eng. 4, 10–13 (2014).
19. Mikolajczyk, K., Zisserman, A., Schmid, C.: Shape recognition with edge-based feactures. Brit. Mach. Vis. Conf., Norwich. 1–10 (2003).