Active object detection using colour

Piotr Jasiobedzki[1], Brian Down[1], James R. R. Service[2], Victor Wu[1]
[1] Dept. of Computer Science, University of Toronto, Canada M5S 1A4
[2] Ontario Hydro Technologies, Toronto, Canada

Abstract

We describe research on active object detection using a robotic head equipped with an imaging / range sensor. This research is part of an ARK (Autonomous Robot in a Known Environment) project to develop novel techniques for autonomous mobile robots. The robot's tasks include detecting landmarks for navigation (their position is known in advance and is used to register the robot on a map) and searching for known objects. The robotic head used in the project is equipped with a colour camera with motorised zoom and a spot laser rangefinder. We present an active technique for detecting objects based on their two dimensional views. The technique relies on using colour to detect possible candidates in a scene and verification of the candidates using their colour, shape and size. We use the head's abilities to point, zoom, focus the sensor and to measure range during the active exploration of the environment. We describe the training and real–time implementation of this process.

1. Introduction

The research described in this paper is part of an ARK (Autonomous Robot in a Known Environment) project to develop techniques for an autonomous mobile robot. The project is a collaborative effort of University of Toronto, York University, Ontario Hydro and Atomic Energy of Canada Ltd [10, 11]. The goal of the project is to build a mobile robot capable of autonomous navigation in a known industrial environment. The robot relies on vision as its main sensor for global navigation. The ARK robot uses a map with permanent objects in the environment (walls, pillars) to plan its path. While executing the planned path, the robot searches the environment for known landmarks. Positions and salient descriptions of the landmarks are known in advance and are stored in the map [7]. The robot uses the relative position of the detected landmark to update its position. Modification of the environment to include unique and easily identifiable beacons is not permitted and the robot uses naturally occurring objects as landmarks.

The industrial environment, for example, industrial bays or power plants are fairly large and require sensors that can operate at distances of tens of meters (figure 1).

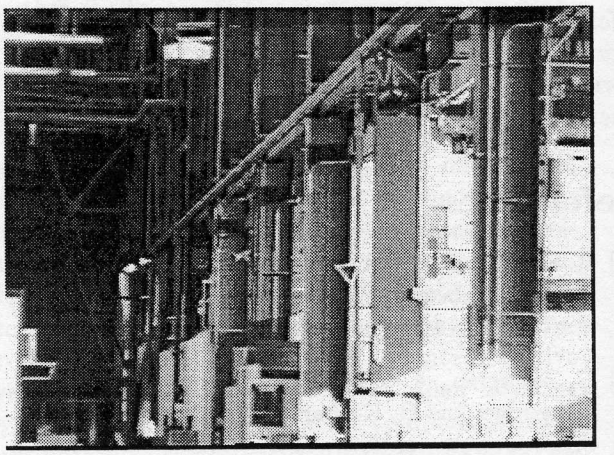

Figure 1. Industrial bay (intensity)

The number of visual features (lines, regions) can be very high and usually exceeds the number of similar features found in office environments. The robot's tasks include detection of landmarks and search for known objects. The robotic head used in the project is equipped with a colour camera, a wide angle, motorised zoom lens and a spot laser rangefinder. We describe research on active object detection using their two dimensional views and utilizing the robotic head. The technique relies on the use of colour to detect possible candidates and to verify the candidates by comparing their colour, shape and size to the known stored attributes of the objects we wish to detect. We enhance detection and simplify verification by using the head's ability to point, zoom, focus the sensor and measure range

2. Active object detection

Visual searching for objects requires scanning the environment or checking expected locations with a camera or even moving a robot. In typical tasks of detecting visual landmarks or searching for a target object, the object itself and its salient characteristics are known in advance. When searching for a landmark the robot can predict where to point the camera as it knows its own approximate location on the map, and the coordinates and pose of the landmark. Still, the uncertainty of its position requires the robot to select a fairly wide field of view for the camera. When searching for an object, the robot has

far fewer constraints on possible locations of the objects ("on the floor", "in area A of the bay"). An attention mechanism that selects some "interesting" locations in an image or environment significantly speeds up and simplifies the search. Features such as intensity, colour, motion and presence of significant edges are often used to focus attention. Once candidate locations have been selected, each of them is inspected closely to verify presence of the target object. This detailed active examination of detected candidates involves directing the camera at the object, changing zoom and focus, and even moving the robot. The estimate of the robot's location and information stored in the map permits selection of an expected view of the landmark. The landmark recognition process can rely on two dimensional images.

The candidate detection has to be performed in close to real–time to allow for rapid localisation. Eventually this will happen while the robot is in motion. The candidate detection scheme should be biased towards detection of false positives, as the verification step will reject artifacts.

We propose using colour to identify possible candidates in an image. Our colour classification scheme consists of an off–line training phase and an on–line classification of pixels on a real–time image processor. Colour information is used for pixelwise classification of images and assigning pixels to possible target candidates or background classes. We achieve the real–time performance by creating look up tables (LUTs) during the training phase and fast indexing during the on–line classification.

The ARK robot is equipped with a special sensor (Laser Eye) that can provide colour images and range data at distances up to 100 m. Such distances are typical for the industrial environment. The Laser Eye is a combined range / video sensor consisting of a camera and a laser range–finder [5, 6]. The range–finder uses the time–of–flight principle and provides a single depth measurement for each orientation of the sensor. Measuring distances to objects in the scene requires pointing the sensor at each of them in turn and reading their depth. The sensor is mounted on a robotic head with four degrees of freedom: two extrinsic — head pan and tilt, and two intrinsic — camera zoom and focus (figure 2). The range–finder uses an infra–red laser diode to generate a sequence of optical pulses that are reflected from a target located up to 100 m away. The time required to travel to and from the target is measured to estimate the distance. The head can move at velocities up to 180 degrees per second.

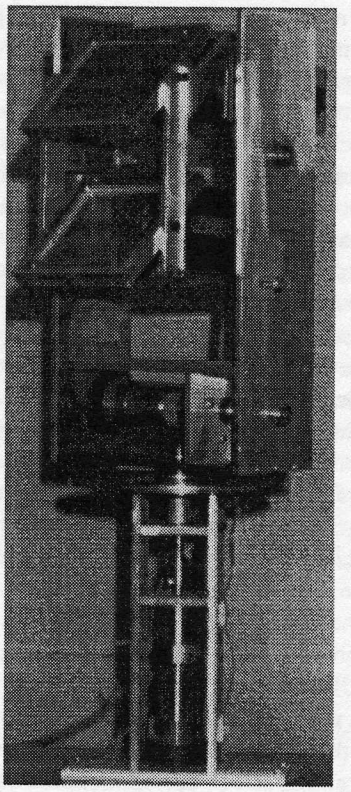

Figure 2. The robotic head used in experiments

3. Use of colour

Analysis of colour images can provide more information and enable fuller description of scenes than analysis of monochrome images. Techniques developed for segmenting monochrome images, such as edge detection and region based segmentation, have been extended to work on multi–dimensional data.

In this section we describe several approaches to rapid colour classification, point out their limitations and then discuss several issues of colour based classification (colour spaces, pattern recognition techniques). The detailed description of our almost real–time colour detection algorithm, that depends on pattern recognition methods, follows in section 4.

3.1. Real–time classification of colour images

Our application requires real–time (or fast — under 1 sec) classification of colour images. Such a speed can be achieved only by comparing each pixel, described by a vector of colour components red, green and blue (RGB), against a set of thresholds or using look up tables (LUT). Using independent thresholds on each colour channel severely limits the ability to partition the colour space. Combining the multiple data streams into one channel permits indexing into the LUT to achieve the real–time performance of an arbitrary (non–linear) conversion. The nature of this conversion is determined by the contents of the LUT. The problem is how to create a LUT that will effectively capture the important variability of the data.

The resolution of the feature space can reach 2^{24} elements (3 x 8 bit colour bands). In many cases it is sufficient to operate on much smaller arrays. For example, Da-

tacube's MV20 advanced processor (AP) uses a look up table with maximum of 64 k entries [2]. The contents of the look up tables are often determined by manual selection. A more systematic approach uses training by showing examples and manually delineating objects of interest in the training images. Cells in colour space corresponding to the feature combinations present in the training set are assigned to appropriate classes. For a low resolution of feature space (several hundred cells) such a technique is sufficient, as camera noise and blur create dense clusters [16]. For high resolution look up tables containing, for example 64 k cells, this approach is not well behaved as insufficient training data creates "holes" in the feature space. Such holes cause misclassification of the data, so various techniques that fill them in and create compact clusters have been developed. This problem is also known as colour generalisation. Massen [9] used morphological closing of clusters represented in a two dimensional space: hue and saturation. Soille [15] applied the grey watershed transform to partition the same space. Batchelor [1] proposed several heuristic algorithms to partition the RGB space.

The problem with all these techniques is the subjectivity in selecting the algorithm that generalises the limited data from the training set. In our opinion, the data itself should define the size and shape of the clusters. We propose to use pattern recognition techniques to create a representation of the training data and to partition the discrete colour space (LUT). In this paper we describe a statistical pattern recognition technique. Other classification techniques, for example, neural networks, fuzzy clustering and classification, and mixture models can be used in a similar way.

3.2. Colour spaces

The basic colour space is defined by the red, green and blue (RGB) colour components of the tristimulus model [13]. Many researchers used this space, although the individual colour components are highly correlated. For example, any change in the illumination level causes simultaneous changes in all three channels. This effect can be reduced by normalising each colour (R, G and B) by their sum (R+G+B).

The basic colour space can be transformed by linear or non–linear transforms into other spaces with more advantageous properties. Linear transforms often suffer from the same problem as the basic space, i.e., correlation between the channels, and their components are non–intuitive. Non–linear transforms give results that are closer to human perception of colour and separate the effect of the illuminance level. A potential problem is in existence of singularities introduced by the non–linear transform. The most popular non–linear transform converts the input data into hue, saturation and intensity space (HSI). Hue of an object corresponds to its colour appearance, saturation indicates the amount of a particular hue perceived in the object, intensity corresponds to brightness. There are various formulation of the HSI space, in our research we use the following one:

$$I = \frac{R + G + B}{3} \tag{1}$$

$$S = 1 - \frac{\min(R, G, B)}{R + G + B} \tag{2}$$

$$H = \text{atan}\left[\frac{3 * (G - B)}{(R - G) + (R - B)}\right] \tag{3}$$

It can be seen, from the above equations, that a singularity exists around R=G=B= 0 and H and S have to be set to some arbitrary values. A small change in any of the RGB components results in large change in hue and saturation. Intensity changes from 0 to a maximum positive value, saturation is between 0 and 1. Hue is a cyclical function and, if represented as an angle 0 – 360 degree, then the hue values of 1 and 359 are close together.

3.3. Segmentation into regions

Segmentation of multi–dimensional data can be carried out by projecting the data into a space of lower dimensionality to reduce the computational requirements or by grouping pixels of similar colour into clusters in the multi–dimensional space. Ohta [12] used multiple one dimensional histograms to segment a colour image in a split–and–merge scheme. The presence of multiple peaks in the histograms indicated that such a region should be further divided. Adjacent regions of similar histograms were merged. Wright used Markov random fields to fuse RGB images [17]. The technique described in this paper relies on clustering and classification in colour spaces — the following paragraphs describe this approach in greater detail.

Clustering of n input vectors (patterns) requires partitioning the space into K clusters $\{C_1, C_2, ..., C_K\}$. Cluster C_k contains n_k vectors and each vector belongs to one cluster only [3, 4]. The clustering is performed so as to minimise the within–cluster variation and maximise the inter–cluster variation. The common measure is the squared Euclidean distance d_k^2 between each vector $\mathbf{x}^{(k)}$ in C_k and the cluster centre $\mathbf{m}^{(k)}$:

$$d_k^2 = \sum_{i=1}^{n_k} (\boldsymbol{x}_i^{(k)} - \boldsymbol{m}^{(k)})^T (\boldsymbol{x}_i^{(k)} - \boldsymbol{m}^{(k)}) \tag{4}$$

where

$$\boldsymbol{m}^{(k)} = \frac{1}{n_K} \sum_{i=1}^{n_K} \boldsymbol{x}_i^{(k)} \tag{5}$$

The optimised criterion is the sum of squared errors for all clusters C_k. The resulting clusters are described by coordinates of their centroids $\mathbf{m}^{(k)}$ and their variance and form hyperellipsoids in the colour space.

Clustering can be used to segment images directly or to represent, in a compact form, variability of the input data for classification of subsequent images. Segmentation of images by direct clustering is computationally expensive and consumes in order of tens to hundreds of minutes on a general purpose workstation.

3.4. Classification of multi–dimensional images

It is often the case, that it is not required to segment a new image each time but only to process every pixel in the new image and assign it to one of the classes defined during training. Various statistical pattern recognition techniques are available [3], for example, the k–nearest neighbour classifier could be used, if all the training data is stored. It is more efficient to store a representation for each class and use it for classification. This representation depends on the distribution model of the data. For example, assuming multivariate normal distributions of clusters in the colour space and equal *a priori* probabilities for each cluster, the Bayes discriminant function [3] can be used:

$$g^{(k)}(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{m}^{(k)})^T cov_k^{-1}(\boldsymbol{x} - \boldsymbol{m}^{(k)}) - \log|cov_k| \qquad (6)$$

where
$\mathbf{m}^{(k)}$ is vector of mean (coordinates of the centroid centre)
cov_k is the covariance matrix

The classification of the image is carried out by calculating the discriminant function $g^{(k)}(\mathbf{x})$ for every pixel described by a vector $\mathbf{x}$ and cluster k. The pixel is assigned a class that includes the cluster of the minimum value of the discriminant function. The classes may contain several clusters. Computational complexity of this technique depends on the number of clusters and the resolution of the LUT. On a standard workstation with 20 clusters and a LUT with 64k cells it is in of order of minutes.

4. LUT — training and classification

The use of look up tables permits real–time classification of colour images. The sections below describe our approach to clustering the input data, training the statistical classifier, creating and using the look up tables.

4.1 Training the classifier

We use classical statistical pattern recognition techniques for the classification of pixels. The training sets consist of images with objects of interest in their natural environment and under different illumination. Each of the pixels in the training set is described by its three colour components: hue, saturation and intensity, obtained by converting images from the RGB space. The training data is processed by a clustering programme that partitions the three dimensional space into clusters of points (figure 3). In the experiments described here we used the K–means clustering algorithm [4]. Clustering using the K–means algorithm is computationally expensive and the result may depend on the selection of seeding values. To accelerate the clustering we resample the original image and create reduced images (1/16 and 1/4). The clustering is performed on the smallest image first and starts from random seed points. Then the cluster centres obtained from lower resolution images are used to seed the clustering at the higher resolution. We achieve reasonable performance (the process takes several minutes) and a good partitioning of the data. Selection of the number of clusters is a difficult problem [4]. The user specifies the number of clusters and receives the result of clustering in the form of segmented training images.

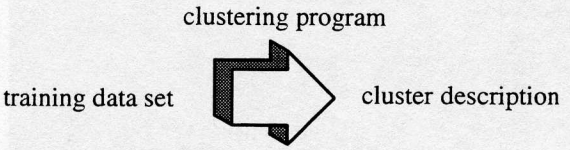

Figure 3. Clustering in a feature space using a training image

After clustering the user assigns individual clusters to classes corresponding to the trained objects or to the background. We have observed a low correlation between the HSI components for the initial set of data. This allowed us to avoid the need to compute and invert the covariance matrix for every input vector $\mathbf{x}$ and the cluster C_k for the quadratic discriminant function (section 3.4). The covariance matrix cov_k for every cluster is assumed to be diagonal and its values are estimated from the intra–cluster variability. The classification programme, described in section 3.4, uses the description of clusters and their class assignment to process all the pixels in a test image. The test image contains all the feature combinations for a given resolution of the input data. The resulting LUT has all its cells filled by this process (figure 4). The resolution of the LUT is limited by the image processing hardware and typically does not exceed 64k bytes. Decomposition of the 24 bit input data (3 x 8 bits of RGB) into 16 bits can be constant for all experiments or may vary depending on the distribution of data in the feature space. Creation of the look up table takes several minutes on a workstation.

4.2 Classification

The on–line classification is performed by combining the colour components of every pixel into one index used

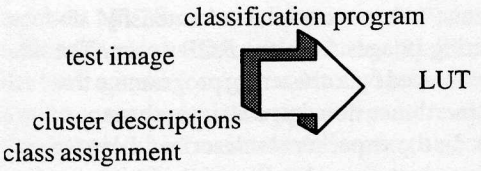

Figure 4. Generation of a LUT using the classifier

to address an entry in the look up table. This entry contains the label corresponding to one of the trained classes (figure 5). The input vectors can be represented in the HSI or in the RGB space. In the first case the look up table performs only classification, in the second it also performs an indirect colour space conversion. This process can be performed on a host taking several seconds for an image or on a specialised hardware where it is carried out in real–time.

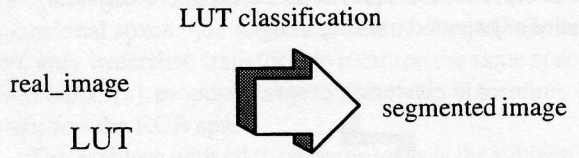

Figure 5. real–time classification using the LUT.

4.3 Candidate selection

Training of the classifier on images obtained under different illumination creates large clusters corresponding to objects. This, together with camera noise, can often produce spurious classification responses as illustrated in section 6, in figure 8. We remove small blobs from the segmented binary images using morphological erosion [14]. The remaining large blobs are reconstructed to be close to their original shape and size by morphological geodesic dilation [8]. The equation (1) defines the reconstruction process: M is the input binary image, B the structuring element.

$$Y = (M \ominus B) \oplus_M B \qquad (7)$$

The following equations define standard dilation and erosion of the set X by the structuring element B. The geodesic dilation of the set X by the element B is performed within the set M.

$$dilation : (X \oplus B)(x) = \{x : B_x \cap X \neq \emptyset\} \qquad (8)$$

$$erosion : (X \ominus B)(x) = \left\{x : \bigcap_{b \in B} X_b\right\} \qquad (9)$$

$$geodesic\ dilation : (X \oplus_M B)(x) = \{x \in M : B_x \cup X \neq \emptyset\} \qquad (10)$$

5. Implementation

We have implemented the training phase (clustering and creation of the LUT) on a Unix host. Training requires some user interaction in selecting images of objects for training and reviewing the clustering the results. The training time depends of the size of the training data set and usually requires minutes on a workstation. This time is not critical as is it is performed off–line. The colour classification has been implemented on the MaxVideo 20 image processing system and is performed in real–time. The data flow in the system is shown in figure 6.

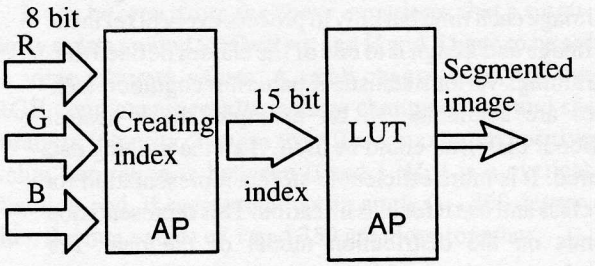

Figure 6. real–time pixelwise classification using the MaxVideo 20

The MaxVideo 20 does support limited morphological operations on the image [2]. Their AP provides a LUT as well as a FIFO that will permit binary morphology on a 3x3 kernel only. Operations with larger structuring elements have to be performed by decomposition of the element and iterating with small structuring elements.

For our application, we would prefer to use a larger kernel, so we have implemented the required binary morphological operations using the convolution and the thresholding operations supported by the MaxVideo 20. Both the convolution and the thresholding are performed in one pass. The following equations define the dilation of a binary image F by a structuring element B. The MaxVideo 20 represents binary images in the same way as grey level images so the input set X ($X \subset F$) is actually represented as a function $f(x)$ defined over the image. The maximum size of the structuring element B is 8 x 8 or 64 x 1 that correspond to the maximum size of the convolution kernel.

$$dilation : g(x) = (f \oplus B)(x) \qquad (11)$$

$$h(x) = \int f(x)B(x - t)dt \qquad (12)$$

$$g(x) = \begin{cases} 1 \text{ if } h(x) > 0 \\ 0 \text{ if } h(x) = 0 \end{cases} \qquad (13)$$

The binary erosion is calculated in a similar way but first the image to be eroded $f(x)$ is complemented:

$$erosion : g(x) = (f \ominus B)(x) \quad (14)$$

$$f_1(x) = \sim f(x) \quad (15)$$

$$h(x) = \int f_1(x)B(x - t)dt \quad (16)$$

$$g(x) = \begin{cases} 1 \text{ if } h(x) > 0 \\ 0 \text{ if } h(x) = 0 \end{cases} \quad (17)$$

The geodesic dilation is used to reconstruct the original shape and size of a blob in the image. This operation requires two input images: a marker image $f(x)$ and the original image $m(x)$. The full reconstruction is performed until idempotence. For practical reasons we implement a limited number of iterations of the elementary dilation step.

$$geodesic \ \ dilation : g(x) = (f \oplus_M B)(x) \quad (18)$$

$$h(x) = \int f(x)B(x - t)dt \quad (19)$$

$$g(x) = \begin{cases} m(x) \text{ if } h(x) > 0 \\ 0 \quad \text{if } h(x) = 0 \end{cases} \quad (20)$$

The real–time implementation of the binary morphology with large (64 x 1, 32 x 2, 16 x 4, 8 x 8, etc.) structuring elements is illustrated in figure 7. The morphological reconstruction consists of three iterations: one erosion and two geodesic dilations.

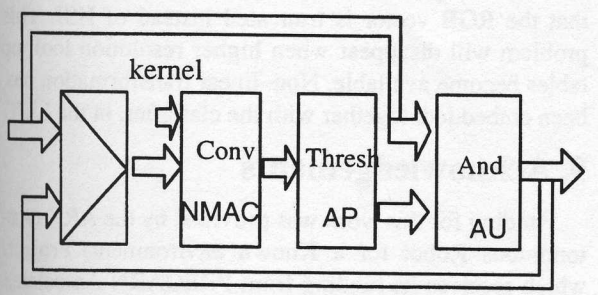

Figure 7. real–time morphology with large structuring elements on the MaxVideo 20

6. Colour classification results

We trained the classifier to detect red and white triangles (fire extinguisher signs) found in several places in the AECL industrial bay (see figure 1). The training set contained multiple triangles hung in various locations throughout the bay. The illumination varied between locations. We used the K–means algorithm to group the data, represented in HSI space, into approximately 10 clusters. The clusters corresponding to inner red triangles were assigned to the sign class, the remaining clusters to the background. The white outline of the sign was excluded from the sign classes as there were many more occurrences of pixels of similar values in this environment. The index to the LUT was composed of the 5 most significant bits of red, green and blue bytes.

Figure 8 shows results of pixelwise classification of the image from figure 1. Each pixel is individually assigned to one of the two classes. As illustrated, the classification result is quite noisy. Some of the red pipes have been picked up by the classifier. There is also a number of isolated pixels mistakenly assigned to the sign class. The image discretisation effect is visible in the shape of the triangular sign.

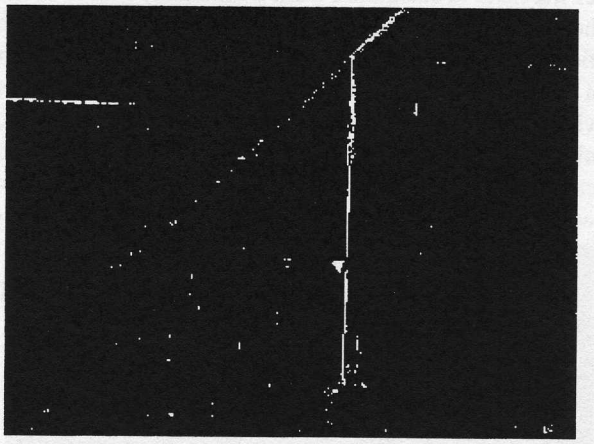

Figure 8. Results of pixelwise classification of data from the image in figure 1

Figure 9 shows the filtered and reconstructed large blobs. The selection of the structuring element size is arbitrary here. This size corresponds to the minimum size of the object in the image coordinates. Only one blob in this image corresponds to the sign and the remaining are artifacts. Determination of whether the detected blobs correspond to the true object or not, could be made at this point,

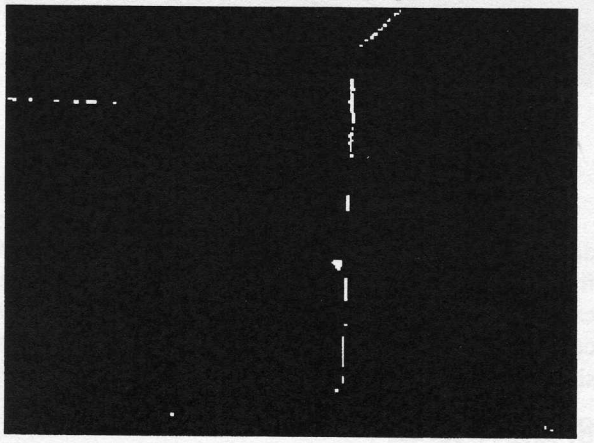

Figure 9. Selected candidates

using a number of different techniques. One such a method would involve calculating shape properties and matching this metric with the expected values. At this resolution, however, it might be difficult to decide if the shape deformations are caused by noise, particularly if the sensor is positioned at a difficult viewing angle. It is much better to point the robotic head at each of these candidates in turn and then acquire and process this new set of images.

7. Active verification of candidates

Each detected candidate is described by a set of parameters that define its position in the image, and the size and location of its bounding window. The new orientation of the head is calculated from a kinematic model of the head that includes the pan, tilt and the initial size of the field of view. The new setting for zoom is selected in such a way that the blob of interest is fully included in the new view but dominates the field of view. If it is not possible to obtain a better view of the candidate by modifying the camera and head settings the platform has to be moved towards the candidate.

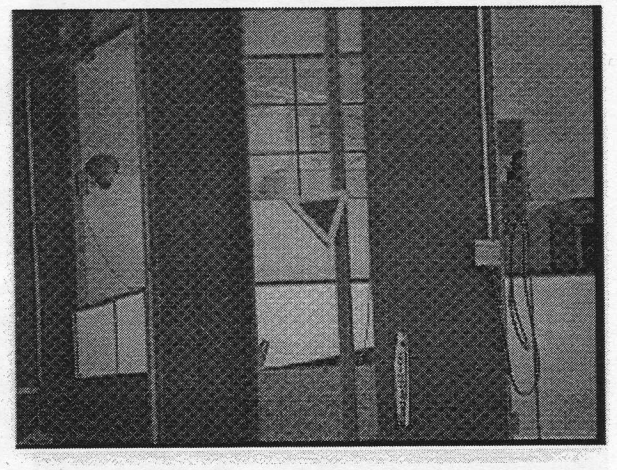

Figure 10. Close up of one of the candidates (intensity)

Figure 10 shows a close up view of one of the candidates. The classified and processed image is displayed in figure 11. The shape parameters of the object can be calculated from this image. The actual size of the object is calculated using the range measurement to the object, its size in image coordinates and the size of the field of view.

8. Discussion

We have described a method of detecting objects using their two dimensional views. The method initially uses colour to select candidates and then their size and shape for final verification. We use a robotic head with four degrees of freedom. The head is equipped with a sensor that provides colour images of the scene and sparse range

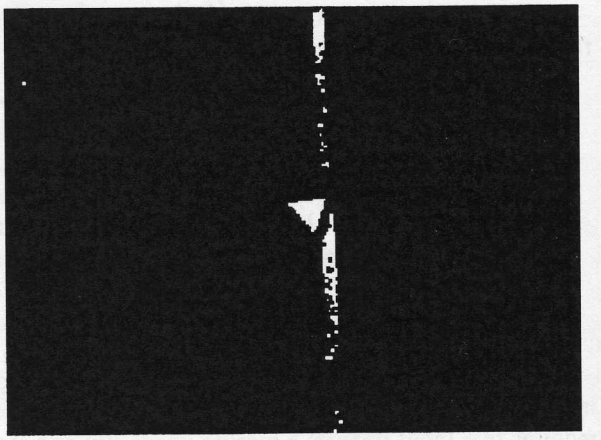

Figure 11. Segmented image from figure 10

measurements at distances up to 100 m. Such distances are typical for the industrial environment of the ARK robot. The colour classifier, described in the paper, has been trained on images of objects in their environment under natural illumination. We are implementing the real–time version of the classifier and the detection process on the MaxVideo 20 vision system (Datacube). The results presented in this paper were obtained for the fire extinguisher signs found in a real industrial setting — the AECL bay. Other landmarks, present in this environment, that can be detected using colour include: doors, area signs and pipes.

One of the limitations of the current implementation is caused by the fact that the hardware used in the experiments has lookup up tables smaller than the maximum number of input vector combinations. Due to the internal architecture, the index for the LUT has to be created before a colour space conversion can take place. This means that the RGB vector is truncated instead of HSI. This problem will disappear when higher resolution look up tables become available. Non–linear transformation has been embedded, together with the classifier, in the LUT.

9. Acknowledgements

Funding for this work was provided by the ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., the Department of Industry, Science and Technology Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro, and Atomic Energy of Canada Limited.

10. References

1. Batchelor B.G.: **Generalisation Procedures for Colour Recognition.** Machine Vision Applications, Architecture and System Integration, Proc of SPIE v. 2064, 1993, pp. 36 – 46.
2. Datacube Inc. **MaxVideo 20.** 1992.

3. Duda R., Hart P.: **Pattern Recognition and Scene Analysis.** J. Wiley 1973

4. Jain A.K., Dubes R.C.: **Algorithms for Clustering Data.** Prentice Hall, 1988

5. Jasiobedzki P.: **Active Image Segmentation using a Camera and a Range–finder.** Applications of Artificial Intelligence XI: Machine Vision & Robotics. Orlando, Florida, April 1993, p. 92 – 99.

6. Jasiobedzki P., Jenkin M., Milios E., Down B., Tsotsos J., Campbell T.: **Laser Eye – a new 3D sensor for active vision.** Intelligent Robotics and Computer Vision: Sensor Fusion VI, Proc of SPIE, vol. 2059, Boston, Sept. 1993, pp. 316 – 321.

7. Jenkin M., Milios E., Jasiobedzki P., Bains N., Tran K.: **Global Navigation for ARK.** Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'93, Yokohama, Japan, July 26–30, 1993, pp. 2165–2171.

8. Lantuejoul C, Maisonneuve F.: **Geodesic Methods in Quantitative Image Analysis.** Pattern Recognition, no 2, vol 17, 1984, pp. 177 – 187.

9. Massen R., Volk G.: **Real–time colour classification for preprocessing photogrammetry images**. SPIE vol. 1395,Close–Range Photogrammetry Meets Machine Vision, pp. 283 – 290.

10. Nickerson B., Jasiobedzki P., Jenkin M., Milios E., Down B., Tsotsos J., Bains N., Wilkes D..: **ARK – Autonomous Mobile Robot in an Industrial Environment.** CIRFFSS '94, AIAA/NASA Conference, Houston, TX, March 21–24, 1994, pp. 12 – 20.

11. Nickerson B., Jenkin M., Milios E., Down B., Jasiobedzki P., Tsotsos J., Bains N., Tran K.: **ARK – Autonomous Navigation of a Mobile Robot in a Known Environment.** Proc. of International Conference on Intelligent Autonomous Systems: IAS–3, Pittsburgh, PA, February 1993, pp. 288 – 296.

12. Ohta Y: **Knowledge–based Interpretation of Outdoor Natural Scenes**. Research Notes in AI 4, Pitman 1985.

13. Pratt W.: **Digital Image Processing.** J Wiley, 1991

14. Serra J.: **Image Analysis and Mathematical Morphology.** Academic Press, 1982.

15. Soille P., Rivest J.F.: **Dimensionality of morphological operators and cluster analysis.** Image Algebra and Morphological Image Processing IV, Proc of SPIE v. 2030, pp. 43 – 53.

16. Swain M., Ballard D.: **Color Indexing**. IJCV 7:1, pp. 11–32.

17. Wright W.A.: **A Markov random field approach to data fusion and colour segmentation.** Image and Vision Computing, vol.7 no 2, 1989, pp. 144 – 150.