Creating Composite Images for Estimating the Effectiveness of Mobile Robot Coverage Algorithms

Sylvia C. Wong[1] Lee Middleton[2]
Bruce A. MacDonald[3]

[1,3] Dept of Electrical & Electronic Engineering, The University of Auckland, New Zealand
[2] ISIS, Dept of Electronics & Computer Science, University of Southampton, United Kingdom

[1,3] {s.wong, b.macdonald} at auckland.ac.nz
[2] ljm at ecs.soton.ac.uk

Abstract

Complete coverage navigation is used in a variety of robot tasks such as vacuuming, surface coating, and systematic foraging. Performance measures are often neglected in this area of research. In our previous work, we have proposed two performance metrics for this type of navigation task and described how these metrics can be estimated from data collected during real robot experiments. Both measures require the estimation of percentage coverage. This paper presents an improved method for extracting the location of the robot from a sequence of captured images, using evidence gathering with a structural model of the robot. The locations extracted are used to create a composite image for use in estimating amount of coverage. The composite images created with evidence gathering are more accurate and tolerant to poor lighting conditions compared to our previous method. Although the new method requires a significant increase in computation time, the time needed is found to be acceptable for evaluating experimental work.

1 Introduction

In tasks such as vacuum cleaning, painting and landmine searching, a mobile robot must visit all reachable floor surface in an enclosed region. This type of behaviour can be described as complete coverage navigation. Complete coverage is different from exploration. In exploration, a robot's sensors covers all of its environment for map building purposes; while in complete coverage, the robot or a coverage tool has to pass over all floor surface.

It is important to measure how well an algorithm performs in real robot experiments. Generally, the measurement of performance is well studied in the area of localisation [Duckett and Nehmzow, 1998, Nehmzow, 2001]. However, it is often neglected in discussions of research on robot coverage navigation. For example, Acar showed only pictures with the route taken by the robot [Acar, 2002]. Ulrich et al. distributed sawdust uniformly on the floor, and measured percentage coverage by the amount of sawdust left afterwards [Ulrich *et al.*, 1997]. Butler used an efficiency measure, coverage factor, which depends on the path length travelled [Butler, 2000]. However, coverage factor does not account for the percentage covered. A robot that repeatedly travelled the same area may have a good coverage factor even though very little of the environment was actually covered.

In [Wong *et al.*, 2002], we introduced two performance metrics for real coverage robots: the effectiveness and the efficiency. Effectiveness is the percent covered and is calculated from a pixel count of a composite image of the coverage process. Efficiency is a measure of unnecessary repetition of robot movements, and is calculated as a normalised path length scaled by the percent covered. We also discussed how these measures can be calculated from data collected during the experiments. Both metrics require the calculation of percentage coverage.

In this paper, we present an improved method for extracting the position of the robot from each frame of a sequence of images. This improvement creates cleaner and more accurate composite maps. As a result, the estimate of percentage coverage is improved. Also this new method is more generic and can be used in situations where only part of the robot is used as a coverage tool.

Note that this paper does not deal with calculating the performance metrics for experiments with *simulated* robots. In simulations, percentage coverage is simply calculated as number of grid cells covered divided by total number of grid cells. Efficiency can be measured with the number of repeatedly covered grid cells [Gabriely and Rimon, 2002, Zelinsky *et al.*, 1993]. This is because repeatedly covering the same grid cell is undesirable and a good al-

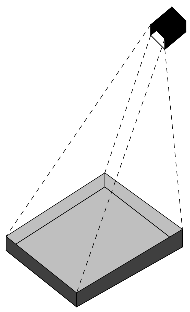


Figure 1: Mounted camera capturing frames of robot's movement.

gorithm would thus minimise the amount of repeated coverage.

Section 2 presents the two metrics used for measuring performance of coverage algorithms. Section 3 explains how evidence gathering is used to extract location of robot from individual frames for creating composite coverage maps. Section 4 compares results from evidence gathering with our previous method, image subtraction.

2 Performance Metrics

The effectiveness of a coverage algorithm is measured as the amount of coverage C

$$C = \frac{\text{Area covered}}{\text{Area of reachable surface}} \tag{1}$$

To estimate the area covered by the robot, a wall mounted camera was used to capture a movie of the robot's progress. A suitable setup is shown in figure 1. The trail of the robot is reconstructed by superimposing the position of the robot in each frame. With this composite image, areas visited by the robot and areas occupied by obstacles can both be estimated by counting pixels.

Unlike in simulation, efficiency cannot be estimated by the number of repeatedly covered pixels in the composite image. This is because a real robot does not move in a grid. It "re-visits" a significantly amount of pixels even when moving forward. Therefore, we measure efficiency as the distance the robot travels instead. It is measured as the comparison of the actual path taken, P_a, with the optimal path. However, Arkin et al have shown that finding the shortest coverage path is NP-hard [Arkin *et al.*, 2000]. As a result, P_a is compared with the idealised minimal path, P_m, instead. The idealised minimal path is the shortest coverage path for a mobile robot that can teleport with no cost associated with the teleport operation. All environments can be covered by such a robot with no retracing. P_m is therefore always equal to or shorter than the realisable shortest path. In summary, the comparison of path length is calculated as:

$$L = \frac{\| P_a \|}{\| P_m \|} \tag{2}$$

However, equation 2 does not take into account the amount of coverage. P_m is the idealised minimal path for covering the entire environment. If an algorithm covers only 50% of a given environment, then P_a should be compared with 50% of P_m instead. A solution to this problem is to scale P_m with C. Thus,

$$L' = \frac{\| P_a \|}{\| P_m \| \times C} \tag{3}$$

In real robot experiments, P_a can be easily obtained from odometry information. Figure 2 shows how P_m is estimated. The free space in an environment is divided into a series of non-overlapping strips. Each strip is the width of the robot (or the width of the coverage tool used). P_m is the total length of these strips.

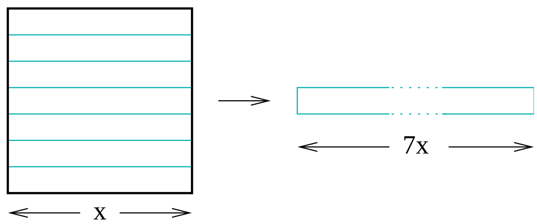


Figure 2: Estimating the idealised minimal path length, $\| P_m \|$.

3 Computing the Composite Coverage Image

There are three major image processing methods that may be employed to create the composite coverage map from a sequence of recorded frames. These methods are *image subtraction*, *template matching*, and *evidence gathering with a structural model of the robot*. Our previous work used image subtraction [Wong *et al.*, 2002]. Image subtraction is efficient and simple to perform, but suffers from two major disadvantages. Firstly, it is not particularly reliable as artifacts frequently appear in the background subtraction process; this is because of environmental factors such as lighting. Additionally, in the case of a tethered robot (such as the khepera, figure 3) the cable motion is computed as part of the final composite image.

The second method, template matching, uses a template [Gonzalez and Woods, 2002] as a representation of the robot and exhaustively searches each frame to find the region which best matches the template. This method suffers from two main disadvantages. Firstly, the search is extremely slow. Secondly, the template does not work very successfully if the robot's shape changes due to perspective or camera distortions.

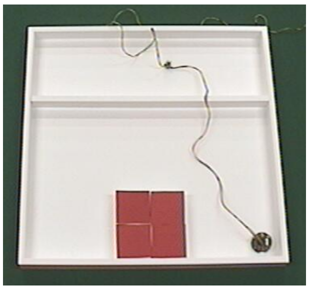


Figure 3: Khepera robot used in our experiments showing the cable.

The third approach, evidence gathering, uses a combination of these two approaches. A model of the robot's shape is devised and a Hough accumulator [Nixon and Aguado, 2002] is used to find the most likely candidate. This approach is faster than template matching and is more robust than pure background subtraction.

3.1 Composite map with Evidence Gathering

Figure 4 describes the required image processing to find the location of the robot in an individual frame. Initially, a frame, f_i, is extracted from a sequence of n images F. The frame is then subtracted from a reference frame r which is an estimate of the images' background. The result of this subtraction, s_i is an image that emphasises the regions in which movement is occurring. The resulting image is then converted to an edge map e_i, by edge detection. The edge map is given as evidence to the model fitting algorithm which uses a Hough transform technique. The result is the coordinates of the centre of the robot and some additional parameters that describe the robot via the model. The pertinent steps will now be discussed in turn.

3.1.1 Background Subtraction

It is desirable that the reference image be an accurate representation of the region without the robot in it. Any error in the reference image will result in artifacts in s_i. Hence, it is desirable that the approach is as robust as possible to changes in illumination within the scene; including both lab and experimental factors. A temporal average is taken of the images in the sequence. A naïve approach would suggest using the average of the images. However, this will result in a ghostlike trail that corresponds to the motion of the moving object. Instead the median operator was employed. This computes the reference image as follows:

$$r(x,y) = \text{median}\Big(f_0(x,y), \cdots, f_{n-1}(x,y)\Big) \qquad (4)$$

In implementation, the median operation is computed via a histogram of image intensities at each point and finding the value corresponding to the 50th percentile. After computation of the reference image the background subtraction process is completed via subtracting the reference image from each of the original images in the sequence. Thus:

$$s_i = f_i - r \qquad (5)$$

As the images employed are full colour (RGB) images then this subtraction is computed for each colour channel separately.

3.1.2 Edge Detection

After the reference image is subtracted, edges are detected to find the edge information for each colour channel. The edge map is computed using a Canny edge detector, for two main reasons. Firstly, it is considered to have an optimal response for step edge responses, and secondly it can serve to reduce noise in the image. This noise reduction is useful as it can help remove small artifacts in s_i due to estimation problems in the reference image. The final edge map, e_i, is computed via weighted summation and thresholding of the individual colour channels. The weighting in the summation allow the emphasis of features in particular colour bands. The value of the threshold is tuned empirically.

3.1.3 Model Fitting

The edge map is then presented as evidence to a model of the robot. The specific model used depends on two factors. Firstly, the shape of the robot. Secondly, how this shape changes because of perspective effects as the robot moves within the environment. For example, the khepera robot was modelled as a circle. This is possible as at the extreme end of the enclosure, where the robot is at the maximum distance from the camera, the khepera is still circular. In this case the model fitting must find the diameter that best fits the khepera in each frame. A Hough transform is used to fit the model to the edge data. The specific model fitting algorithm is shown below:

For a circular fit, the image is initially examined to find all the edge points. For each edge point, all the points are computed that are in a circle of diameter d and centred on this edge point. Diameter was employed instead of radius as it is easier to compute this from an image sequence. If the computed point is another edge point, then an array is incremented with a point indexed by the original edge point and the diameter.

After evidence is gathered from the entire image, the array $A(x, y, d)$, will contain peaks which correspond to likely circle centre points and diameters. Within this array the peak corresponds to the circle centre and

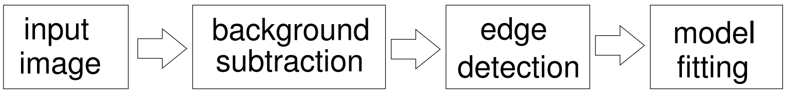


Figure 4: Extraction of the location of the robot in a single frame.

Algorithm 1 Hough Evidence Gathering

for $d = D_{\min}$ to $D_{\max}$ **do**
  for all edge pixels (x, y) **do**
    for $\theta = 0$ to 2π **do**
      $x_c \leftarrow x - \frac{d}{2}\cos\theta$
      $y_c \leftarrow y - \frac{d}{2}\sin\theta$
      if (x_c, y_c) is in image and (x_c, y_c) is an edge point **then**
        increment $A(x, y, d)$
      end if
    end for
  end for
end for

diameter which occurs most often. This is the best candidate for the robots location within the frame. It can be computationally expensive to search all possible diameters so the possible search space is reduced by examination of the original image sequence. The purpose of this is to find estimates of the diameters of the robot at the maximum and minimum distance from the camera. This sets upper and lower bounds to the diameter parameter, $D_{\min}$ and $D_{\max}$.

The output of the model fitting stage will be a series of numbers which describe the central coordinates of the robot and the model chosen to describe the robot. By examining the results from all successive frames an accurate representation of the robot's path can be found. This, along with the model of the robot, can be used to produce a composite image for calculating the amount of coverage C.

4 Results and Discussion

Figure 5 shows composite images created from frames taken on a sunny day with good lighting conditions. Figure 5(b) is created using image subtraction, our previous method; and figure 5(c) is produced with our new evidence gathering method. In this case, the resultant composite images created from the two methods look similar.

Figure 6 shows composite images created from frames taken on a rainy day. With poor lighting conditions, it can be seen that the contrast in the reference image (figure 6(a)) is a lot lower. Figure 6(b) is created with image subtraction, and figure 5(c) is produced with evidence gathering. It is obvious from that the resultant composite image created by our new method is better.

The tiny gaps present in the top part of the tray in figure 6(b) are artifacts from the image subtraction method. Examining the original frames showed that the region was properly covered by the robot. This occurs because the robot extracted using image subtraction sometimes is incomplete. This effect is illustrated in figure 7. Figure 7(a) shows the original frame used. Figure 7(b) shows the result of extraction using image subtraction. The robot is incomplete and does not appear solid. In comparison, evidence gathering does not suffer from this problem, as seen in figure 7(c).

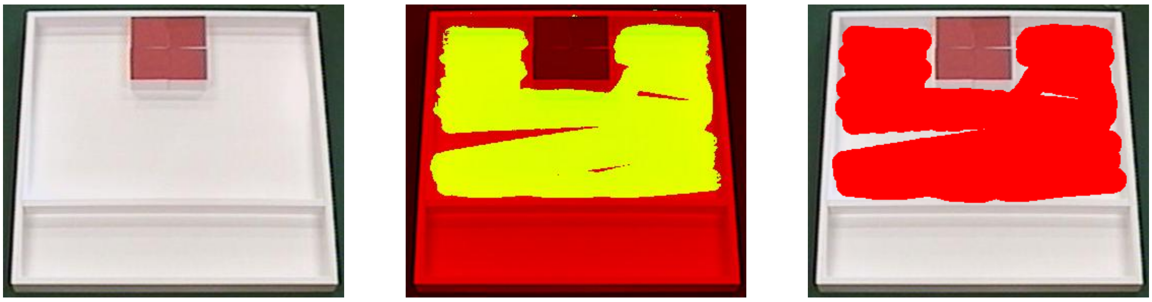
	Figure 5	Figure 6
Image subtraction	91.7%	95.4%
Evidence gathering	95.7%	99.2%

Table 1: C calculated from composite images.

After the perspective warping is corrected [Wolberg, 1990], the amount of coverage C is estimated for each of the composite images by counting pixels. The results are shown in table 1. It can be seen that C calculated from composite images created by the two different methods differs by nearly 4%. This difference is due to the more complete robot silhouette as evidenced in figure 7. Since composite images created with evidence gathering are more accurate than those created with image subtraction, C calculated from the former method is expected to be closer to the real value.

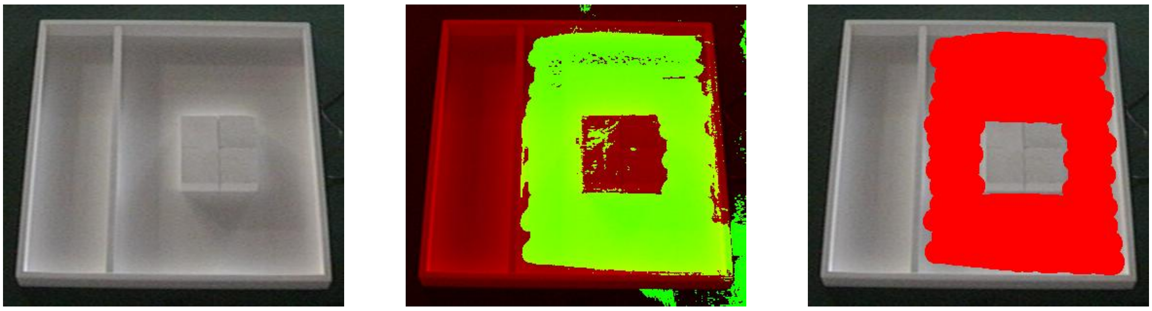
By using a different model, the location of a robot of a different shape can be extracted. Figure 8 shows the centres of the top of a B21r robot extracted using the evidence gathering method and using an ellipse as a model of the B21r. Note that the cylindrical robot does not appear circular because the camera not mounted directly above the environment. To extract an odd shaped robot or a tool carried by a robot, a marker can be used and a model is fitted for the marker instead. With the image subtraction method, the entirety of the robot is extracted and centres are never found. This means that image subtraction cannot be used to create composite maps for coverage with tools that present partial views of the robot. In comparison, evidence gathering can fit a model to find the position of the tool instead of the robot position.

Since the output of the extraction contains a list of centres, the path taken by the robot can also be recreated. An example of this is shown in figure 9. As a result, the path length taken by the robot can be calculated from this list. This can be compared with measurements from odometry information from the robot.



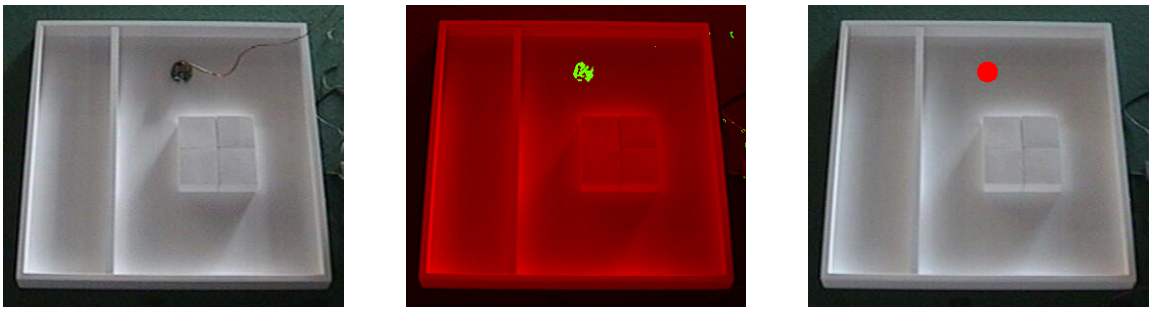
(a) (b) (c)

Figure 5: Frames taken under good lighting condition. (a) Reference image. Composite image created using (b) image subtraction, (c) evidence gathering.



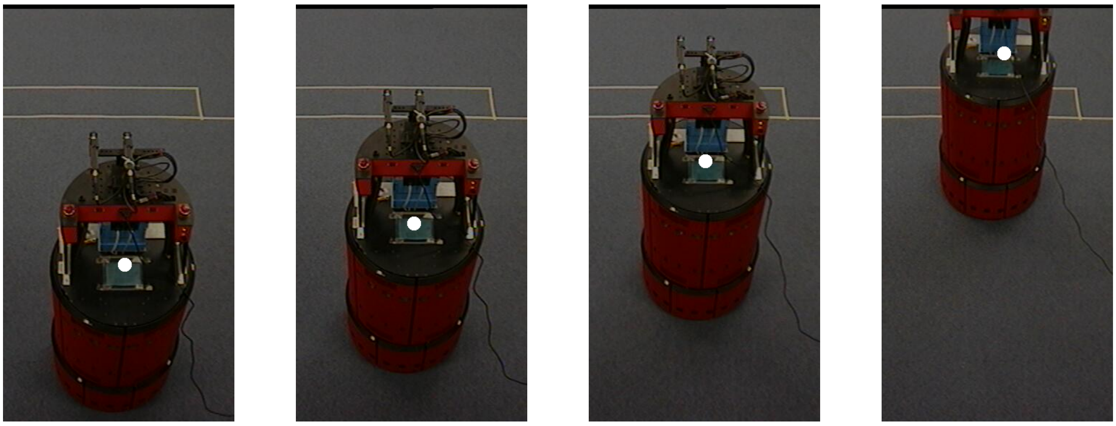
(a) (b) (c)

Figure 6: Frames taken under poor lighting condition. (a) Reference image. Composite image created using (b) image subtraction, (c) evidence gathering.



(a) (b) (c)

Figure 7: (a) Extraction from a single frame. (b) Robot extracted using image subtraction appears "porous". (c) Evidence gathering does not suffer from this effect.



(a) (b) (c) (d)

Figure 8: Extracting the location of a B21r robot using evidence gathering. The white dots correspond to where the centres are found.

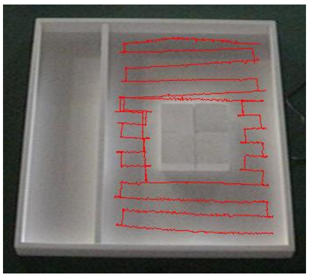


Figure 9: Path taken by robot recreated using locations of centres extracted.

A disadvantage of evidence gathering is its high computational cost compared to the image subtraction method. For a sequence of around 3000 frames, image subtraction takes only a few minutes, while evidence gathering takes several hours. However, since performance metrics are used offline for data analysis, the speed of the evidence gathering method is acceptable as processes can be left running on a computer cluster overnight.

There are two possible ways to improve computation efficiency. First, the current implementation used Python, a scripting language. The speed of computation may be improved by implementation using a compiled language such as C. Secondly, the image size to be analysed can be limited to around the robot's position after the first frame. Since the computation time of the model fitting procedure depends on the number of edge points, not the image size, the improvement in speed cannot be estimated without studying edge pixels distribution in the original frames. However, neither method will make evidence gathering real time.

5 Conclusions

This paper presented an improvement on our previous algorithm for creating a composite coverage image. Since both metrics for efficiency and effectiveness of performance of coverage navigation depend on the amount of coverage achieved, creating an accurate composite image is very important.

Evidence gathering with a structural model of the robot is used for creating composite coverage maps. For each frame in the sequence, a reference background image is first subtracted and edge detected. The resulting edge map is then presented as evidence to a model of the robot. This model fitting finds the central coordinates of the robot and the parameters of the model chosen to describe it. By examining the whole sequence, the path taken by the robot can be re-created to create a composite coverage map.

Compared to image subtraction, the method we previously used, composite images from evidence gathering suffer from less artifacts and are more tolerant to different lightning conditions. It can also be used in experiments where coverage tools are used. Although the computational cost is higher, the increase is deemed acceptable.

References

[Acar, 2002] Ercan Umut Acar. *Complete Sensor-based Coverage of Unknown Spaces: Incremental Construction of Cellular Decompositions.* PhD thesis, Carnegie Mellon University, Pennsylvania, 2002.

[Arkin *et al.*, 2000] Esther M. Arkin, Sandor P. Fekete, and Joseph S. B. Mitchell. Approximation algorithms for lawn mowing and milling. *Computational Geometry*, 17(1-2):25–50, 2000.

[Butler, 2000] Zack J. Butler. *Distributed Coverage of Rectilinear Environments*. PhD thesis, Carnegie Mellon University, Pennsylvania, 2000.

[Duckett and Nehmzow, 1998] Tom Duckett and Ulrich Nehmzow. Mobile robot self-localisation and measurement of performance in middle-scale environments. *Robotics and Autonomous Systems*, 24:57–69, 1998.

[Gabriely and Rimon, 2002] Yoav Gabriely and Elon Rimon. Spiral-STC: An on-line coverage algorithm of grid environments by a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 954–960, Washington, DC, May 2002.

[Gonzalez and Woods, 2002] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, second edition, 2002.

[Nehmzow, 2001] Ulrich Nehmzow. Quantitative analysis of robot-environment interaction - on the difference between simulations and the real thing. In *Proceedings Eurobot*, 2001.

[Nixon and Aguado, 2002] Mark S. Nixon and Alberto S. Aguado. *Feature Extraction & Image Processing*. Butterworth-Heinemann, 2002.

[Ulrich *et al.*, 1997] Iwan R. Ulrich, Francesco Mondada, and J. D. Nicoud. Autonomous vacuum cleaner. *Robotics and Autonomous Systems*, 19(3–4):233–245, March 1997.

[Wolberg, 1990] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.

[Wong *et al.*, 2002] Sylvia C. Wong, Lee Middleton, and Bruce A. MacDonald. Performance metrics for robot coverage tasks. In *Proceedings Australasian Conference on Robotics and Automation (ACRA)*, pages 7–12, Auckland, New Zealand, 2002.

[Zelinsky *et al.*, 1993] A. Zelinsky, R. A. Jarvis, J. C. Byrne, and S Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robots. In *International Conference on Advanced Robotics ICAR*, Tokyo, Japan, November 1993.