

Research Article

Adaptive Colour Feature Identification in Image for Object Tracking

Feng Su,¹ Gu Fang,¹ and Ngai Ming Kwok²

¹ *School of Computing, Engineering, and Mathematics, University of Western Sydney, Locked Bag 1797, Penrith, NSW 2751, Australia*

² *School of Mechanical and Manufacturing Engineering, The University of New South Wales, Sydney, NSW 2052, Australia*

Correspondence should be addressed to Gu Fang, g.fang@uws.edu.au

Received 8 October 2012; Accepted 14 November 2012

Academic Editor: Sheng-yong Chen

Copyright © 2012 Feng Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Identification and tracking of a moving object using computer vision techniques is important in robotic surveillance. In this paper, an adaptive colour filtering method is introduced for identifying and tracking a moving object appearing in image sequences. This filter is capable of automatically identifying the most salient colour feature of the moving object in the image and using this for a robot to track the object. The method enables the selected colour feature to adapt to surrounding condition when it is changed. A method of determining the region of interest of the moving target is also developed for the adaptive colour filter to extract colour information. Experimental results show that by using a camera mounted on a robot, the proposed methods can perform robustly in tracking a randomly moving object using adaptively selected colour features in a crowded environment.

1. Introduction

Surveillance is the task of monitoring the behaviours and/or activities of people from a distance. Security cameras are considered to be the most commonly used equipment. These cameras are used for applications such as, industrial process control, traffic monitoring, and crime prevention. However, despite their wide usages, security cameras still have many weaknesses. One of the weaknesses is its blind spot [1], since they are mounted on mechanical hinges, cameras are only able to monitor at certain angles, and the security system can be infiltrated through those unseen areas. Another weakness rests on the involvement of human operators [2], who usually monitor a large number of inputs from cameras. Because these operators could be subjected to boredom, fatigue, and distractions, it is possible that they

fail to detect criminal or other unwanted behaviours. Therefore, a mobile robot could be used to overcome these potential problems. A robot would be able to travel throughout the monitoring areas autonomously and continuously, making its own decisions while identifying the unwanted behaviours or activities, and respond accordingly such as sending alerts.

Object tracking using computer vision is a crucial component in achieving robotic surveillance. The goal of object tracking is to track the position of the moving objects in a video sequence. This can be achieved by identifying and tracking a specific feature such as colour that belongs to the moving object. The trajectories of the moving object can then be traced through the process over time.

Most existing colour tracking methods are designed to track a fixed salient colour feature. However, if the camera is moving then the tracked colour feature may be no longer salient due to the changing environment. In this case, the tracking may take place to follow a wrong object. Therefore, new methods are required so that the colour feature can be determined adaptively according to the environment the camera is operating in.

The main contribution of this paper is the introduction of a colour filtering method that is capable of adaptively identify the most salient colour feature that belongs to the moving object and using this colour feature for tracking. If the saliency of the chosen colour feature changes due to the operating environment and lighting conditions, the filter will automatically determine a different colour to track. A method of determining the region of interest (ROI) of the moving target is also introduced for the adaptive colour filter to extract colour information.

This paper is organised into six sections. The related research in the area is reviewed in Section 2. The method of determining the ROI is provided in Section 3. In Section 4, an adaptive colour filtering method is introduced. Experimental results are presented in Section 5, and conclusions are given in Section 6.

2. Related Work

2.1. Moving Object Detection

Moving object detection and tracking is an important and fundamental topic in computer vision. Its applications can be found in a large number of engineering fields including: traffic monitoring [3], video surveillance [4], autonomous navigation [5], and robotics [6–10]. One of the most commonly used methods to detect moving objects is background subtraction [11, 12].

Background subtraction involves the separation of moving foreground objects from the static background. The fundamental assumption of the algorithm is that the background is relatively static compared to the foreground. When objects move, the regions in a set of video frames that differ significantly from the background model can be considered to be the foreground (moving objects). A vast amount of research in moving object detection has been done with many algorithms proposed. The most fundamental method uses the Gaussian Mixture Model (GMM) [13]. This method models the evolution of each background pixel intensity by a mixture of K (a small number usually from 3 to 5) Gaussian distributions. There have also been many revised and improved methods based on GMM. One of them is the Improved Adaptive Gaussian Mixture Model (AGMM) [14]. In this model, both the parameters and the number of components of the mixture are constantly adapted. Another

enhanced version is the Improved Adaptive Background Mixture Model (IABMM) [15]. In the work reported, the likelihood factor is removed from the GMM, because it causes slow adaptations in the means and the covariance matrices which can result in failure of the tracker. The IABMM also contains an online Expectation Maximization algorithm which provides a good initial estimate by expected sufficient statistics update equations before enough samples can be collected. Other background subtraction methods include the Codebook [16]. In this method, each pixel is represented by a Codebook which is a compressed form of background model for a long image sequence. This allows the method to capture background variation over a long time with a low memory requirement.

2.2. Moving Object Tracking

Although the background subtraction-based method can robustly identify moving objects with a stationary camera, it cannot provide satisfactory results with a moving camera. This is because the background subtraction methods extract the foreground by distinguishing the differences between the moving objects and a “stationary” background. This difference finding mechanism is built on the assumption that background stays longer and is more static when comparing to the foreground. If a moving camera is used, for example, a camera mounted on a mobile robot, background subtraction will face the problem that the background of the image is constantly changing due to camera movement. This will lead to false classification of the majority of an image to be foreground. This false classification will cause the moving camera system to lose track of the target object. Therefore, existing object tracking methods using mobile robots (moving cameras) usually rely on certain features belong to the tracked objects, such as colour.

A moving object tracking method using mobile robot is performed with background subtraction and colour probability distribution [17]. This is done by stopping the robot when background subtraction is performed and using the colour probability distribution information to track the target. This method assumes that the colour of the tracked object never changes which is not always the case. Also, the locomotion of the robot was remotely controlled in the experiment rather than fully autonomous.

Another object tracking approach is developed based on scale invariant feature transform (SIFT) and mean shift [18]; SIFT is used to find features corresponding to the region of interests, while mean shift is used to find similarities in the colour histograms. This method combines the advantages of both SIFT and mean shift to achieve more accurate tracking results; however, due to high computation costs, it also has the slowest processing speed (1.1 fps) when compared to SIFT or mean shift alone. This level of computation complexity has imposed difficulties in real-time applications.

Object tracking method can also be used for skin segmentation and the tracking of sign language recognition [19]. This method could track the face and the hand accurately using a colour model with a stationary camera. However, the testing background is fairly simple, the distance between the testing subject and the camera remains very close as the subject always occupies large part, sometimes more than half, of the image.

Other colour tracking methods include tracking by transductive learning which requires a high computational cost [20], colour tracking specifically designed for illumination variations in an uncontrolled operating environment [21], multicamera colour tracking which relies on accurate target identification between different cameras [22], kernel-based object tracking using colour and boundary cues [23], object tracking using mean shift with

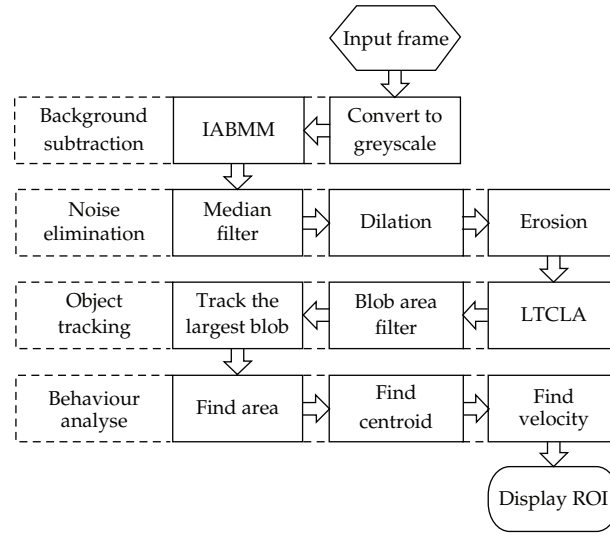


Figure 1: Flowchart of the ROI determination algorithm.

a clustering algorithm and colour model [24], selecting reliable features from colour and shape-texture cues [25], or using area weighted mean of the centroids [26].

The majority of existing colour tracking methods is designed for stationary cameras without the use of mobile robots. Furthermore, the environments presented within these methods are generally stable with little to no variances. To the best of our knowledge, there is no reported method on using GMM-based approach with adaptive colour feature selection in moving object tracking using a robot-mounted camera.

3. Region of Interest (ROI) Determination

As illustrated by the flowchart in Figure 1, the ROI determination algorithm starts by converting the RGB input to both Hue-Saturation-Value (HSV) and greyscale images. It includes four major stages: background subtraction, noise elimination, object tracking, and behaviour analysis. More details of each step are given in the following sections.

3.1. Background Subtraction

To perform background subtraction, the live RGB videos are firstly converted into greyscale images. Greyscale images are used as inputs for background subtraction process because they require less memory to operate and produce faster processing speed than colour images [27, 28].

Then, the IABMM [15] method is used to identify the moving objects and present them in a binary image as foreground objects using white pixels, while allocating all stationary objects as the background using black pixels. The IABMM method is used because it has a faster learning rate and a lower computation requirement than GMM [15, 29]. Thus, it is very efficient to detect motions of objects especially in indoor environments.

The IABMM method used is an improved version of the GMM [13], it begins with the original GMM equation to determine the probability of a given pixel with value X_t at time t using

$$P(X_t) = \sum_{i=1}^K \omega_i \eta(X_t; \mu_i, \Sigma_i), \quad (3.1)$$

where K is the number of Gaussians used for the mixture, ω_i is the weight parameter of the i th Gaussian component, and $\eta(X_t; \mu_i, \Sigma_i)$ is the normal distribution of the i th component with the mean μ_i and the covariance Σ_i .

The major improvement of the IABMM [15] method is the inclusion of the online Expectation Maximization algorithm. This is done by updating the model differently at different phases. Initially when the number of frames t is smaller than L , the model is updated according to the expected sufficient statistics update equations, shown in (3.2), (3.3), and (3.4), then switch to L -recent window update equations when the first L samples (frames) are processed, see (3.5), (3.6), and (3.7). The expected sufficient statistics update equations increase the performance in the beginning while providing a good estimate by allowing fast convergence on a stable background model. The tracker can also adapt to changes in the environment because the L -recent window update equations gives priority over recent data;

$$\omega_{k,t+1} = \omega_{k,t} + \frac{1}{t+1} (P(\omega_k | X_{t+1}) - \omega_{k,t}), \quad (3.2)$$

$$\mu_{k,t+1} = \mu_{k,t} + \frac{P(\omega_k | X_{t+1})}{\sum_{i=1}^{t+1} P(\omega_k | X_i)} (X_{t+1} - \mu_{k,t}), \quad (3.3)$$

$$\Sigma_{k,t+1} = \Sigma_{k,t} + \frac{P(\omega_k | X_{t+1})}{\sum_{i=1}^{t+1} P(\omega_k | X_i)} \left((X_{t+1} - \mu_{k,t})(X_{t+1} - \mu_{k,t})^T - \Sigma_{k,t} \right), \quad (3.4)$$

$$\omega_{k,t+1} = \omega_{k,t} + \frac{1}{L} (P(\omega_k | X_{t+1}) - \omega_{k,t}), \quad (3.5)$$

$$\mu_{k,t+1} = \mu_{k,t} + \frac{1}{L} \left(\frac{P(\omega_k | X_{t+1}) X_{t+1}}{\omega_{k,t+1}} - \mu_{k,t} \right), \quad (3.6)$$

$$\Sigma_{k,t+1} = \Sigma_{k,t} + \frac{1}{L} \left(\frac{P(\omega_k | X_{t+1}) (X_{t+1} - \mu_{k,t})(X_{t+1} - \mu_{k,t})^T}{\omega_{k,t+1}} - \Sigma_{k,t} \right), \quad (3.7)$$

where $\omega_{k,t+1}$, $\mu_{k,t+1}$, and $\Sigma_{k,t+1}$ are the estimates of weight, mean, and covariance of the k th Gaussian component at time $t+1$, respectively. $P(\omega_k | X_{t+1})$ is the posterior probability that X_{t+1} is generated from the k th Gaussian component. Note that $P(\omega_k | X_{t+1}) = 1$ for the matched model and 0 for the remaining models.

3.2. Noise Elimination

After the background subtraction, noise elimination is performed to filter possible noises caused by reflections or motion blurs. The noise elimination consists of median filtering and binary morphological operations.

Median filter [30] is used to remove the so-called “salt and pepper” noise and to restore foreground pixels while preserving useful details.

Noises caused by a changing background or illumination condition may misidentify some of background pixels to be the foreground objects or produce gaps or holes within the foreground objects and separate them into different regions.

Morphological operations, that is, dilation and erosion [31], are used to reduce noise by connecting possible foreground regions and removing any false ones. Dilation is done by first computing the maximal pixel value overlapped by the kernel and then replaces the image pixel under the anchor point with that maximum value. The kernel used is a 3 by 3 kernel with the anchor at its centre. Moreover, erosion is the converse function that operates with minimum value instead of maximum. By combining dilation and erosion, this results in morphological closing which causes the bright regions to join together to form blobs and therefore improves the detection of foreground (represented by white blobs).

3.3. Object Tracking

Object tracking stage begins with the tracking of the blobs (if any) from the output binary image of noise elimination, and they are tracked using linear time component labelling algorithm (LTCLA) [32], which is a fast labelling technique that labels connected components and their contours simultaneously.

The major component of this algorithm involves a contour tracing technique with a tracer to detect the external contour and internal contours of each component. Once a contour point is identified, the tracer works by searching other contour points among its eight neighbours in a clockwise direction. If the initial point is the starting point of an external contour, the search begins at top right, while if the initial points is the starting point of an internal contour, then the search begins at bottom left. Moreover, when the initial point is not the starting point of a contour, the search begins at a point located at 90 degrees clockwise from the position of the previous contour point. This process also marks surrounding pixels (background represented with black pixels) when tracing the contour of the component (foreground represented with white pixels). By marking surrounding background pixels, this ensures no overtracing of the contour occurs. Unless the initial point is identified as an isolated point, the tracer is used continuously in this contour tracing procedure to output the contour point following the initial point until a full loop is done which the entire external or internal contour is traced. LTCLA is used because of its high efficiency. Once the labelling stage is completed, blobs can then be selected for further analysis through a series of filtering processes.

Blobs identified using the LTCLA need to be filtered to eliminate unnecessarily small blobs in order to reduce computational costs. This can be done by using a blob area filter. In our experiments, any blob that contains the pixel number below 0.5% of the total number of pixels in the image is eliminated.

The next step in the object tracking stage is the determination of which blob to track. In this paper, it is decided that the tracking object is the largest moving blob identified. Although the algorithm can be used to track multiple targets, it has been found the speed of the algorithm is heavily influenced by the number of objects tracked. Therefore, in this method, only the largest blob is selected and tracked. This simplification is also justified by the fact that one robot can only track one moving object.

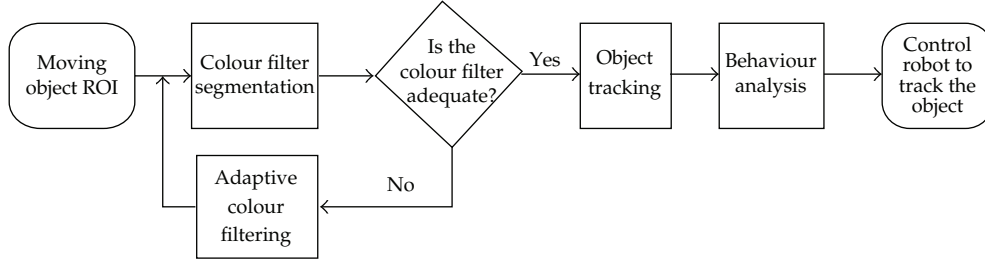


Figure 2: The control concept of the ACF.

3.4. Behaviour Analysis

After the identification of the largest blob, the final stage of the ROI determination algorithm is the behaviour analysis of the blob. The area, centroid, and velocity are obtained and can be used to determine the behaviour characteristics of the object.

To calculate these behaviour characteristics, a ROI is established by a bounding box that encloses the target object and is determined by using the maximum width and height of that object. The area of the object is calculated by counting the number of pixels existed in the tracked blob.

The centroid coordinates X_C and Y_C for the blob and can be found using the centre-of-mass;

$$X_C = \frac{\sum_{i=1}^K X_i}{K}, \quad (3.8)$$

$$Y_C = \frac{\sum_{i=1}^K Y_i}{K},$$

where X_i and Y_i are the x and y location of the i th pixel in the image plane, while K is the total number of pixels belong to the blob.

Once the coordinate of the centroid is found, the velocity can be obtained by comparing the centroid's locations between video frames obtained at different time steps. In this method, previous four centroid coordinates are stored to indicate the path and moving direction of the tracked target, while the velocity is calculated from the difference in pixel locations of the centroid in the current and that of the immediate last image.

4. Adaptive Colour Filter for Moving Object Tracking

The adaptive colour filter (ACF) is a colour tracking method developed in this paper for using a robot mounted camera to track a moving object. The control concept of the ACF is shown in Figure 2.

Initially, the robot and the camera are stationary. Moving objects are detected using the IABMM method. Once the ROI of the moving object is determined, the colour information of the object and the background is filtered using a colour filter to find the most salient feature in the object for tracking. This feature is then used to track the object. When the selected colour feature becomes no longer salient due to changed environment, an adaptive method

Table 1: List of properties defined for the 15 colour filter segments.

Filter segment	Colour	HSV range					
		Hue (°)		Saturation		Value	
		Min	Max	Min	Max	Min	Max
1	White	0	360	0	0.25	0.76	1
2	Grey	0	360	0	0.25	0.26	0.75
3	Black	0	360	0	1	0	0.25
4	Red	0 346	15 360	0.26	1	0.26	1
5	Orange	16	45	0.26	1	0.26	1
6	Yellow	46	75	0.26	1	0.26	1
7	Chartreuse	76	105	0.26	1	0.26	1
8	Green	106	135	0.26	1	0.26	1
9	Spring green	136	165	0.26	1	0.26	1
10	Cyan	166	195	0.26	1	0.26	1
11	Azure	196	225	0.26	1	0.26	1
12	Blue	226	255	0.26	1	0.26	1
13	Violet	256	285	0.26	1	0.26	1
14	Magenta	286	315	0.26	1	0.26	1
15	Rose	316	345	0.26	1	0.26	1

is introduced to update the colour selection. The following sections provide details of this method.

4.1. The Colour Filter

After the ROI is established, the HSV colour space of both the ROI and the entire image are analysed using the proposed colour filter. HSV is selected over RGB because HSV performs better in identifying objects under different lighting conditions such as in shadow, shade, and highlights. This allows the filter to have fewer segments than in RGB [33, 34]. Also, RGB colour space tends to merge neighbouring objects with different colour together and produce blurred results, while since HSV colour space separates out the intensity from the colour information, the results it produced tend to distinguish neighbouring objects with different colour by sharpening the boundaries and retaining the colour information of each pixel [35].

The proposed colour filter is designed to cover the entire HSV colour space. The number of segments of the HSV space can be defined by the user with consideration of hardware limits such as the resolution of the camera and the processing speed required for the algorithm. In this paper, 15 colour filter segments are chosen with their HSV component properties determined empirically, see Table 1.

To apply the colour filtering method, every pixel is checked against the H (hue), S (saturation), and V (value) values for a specific colour filter segment and is regarded as belong to the segment if all H, S, and V values of the pixel fall into the segment range. A uniqueness measure, U , is introduced, shown in (4.1), to measure the ratio of number of

filtered pixels that belong to the foreground (ROI), over the filtered pixels that belong to the background (outside the ROI) in that colour segment;

$$U_i = \frac{F_i}{B_i}, \quad (4.1)$$

where U_i is the uniqueness level of i th colour segment, and F_i and B_i are the number of foreground and background pixels existed in the i th colour segment.

The colour segment with the highest uniqueness value U is picked and applied to the HSV space of the same image. The HSV space is then filtered according to HSV range values of that colour segment (the 15 segments are listed in Table 1).

This filtering process will result in a binary image that has only the pixels with the colour segment of the highest U in the original image to be white while all other pixels to be black. Then, by using object tracking and behaviour analysis, same as in Sections 3.3 and 3.4, the properties required for tracking can be obtained. The tracking process can then be carried out by sending commands to the mobile robot to follow the foreground object.

4.2. Adaptive Colour Filtering (ACF)

An adaptive filtering method is introduced to allow the “picked” colour segment to be re-assigned when the saliency of the tracked colour is no longer prominent or when the target object changed colour appearance possibly due to changing lighting conditions.

To invoke the application of the ACF, one of the following three conditions has to be met.

(1) When a sudden increase of pixels that share the same colour as the tracked target object is detected and exceeds a certain threshold as

$$\frac{B_i}{B_{i-1}} > T_1, \quad (4.2)$$

where B_i is the total number of the detected pixels presented in the i th image frame, T_1 is a control threshold, and the value used is $T_1 = 2$. This threshold value is chosen as it is expected that a 100% increase in the pixel number that share the same colour as the identified colour would make the chosen colour no longer unique or salient. This threshold value, T_1 , was chosen by running the experiments under different conditions in different environments. It was found that by making the T_1 smaller it would increase use of ACF therefore increase the computational load. However, if T_1 is set too large, then the wrong tracking will occur. It was found that by setting the current value of 2 a balanced outcome could be achieved.

(2) When the blob is too large and has covered a significant part of the image, as determined by

$$\frac{(x_{\max} - x_{\min})(y_{\max} - y_{\min})}{X_d Y_d} > T_2, \quad (4.3)$$

where x_{\min} , x_{\max} , y_{\min} , and y_{\max} are the minimum and maximum x and y values of the ROI of the tracked target object, X_d and Y_d are the width and height of the input frame determined

by the camera resolution, T_2 is a threshold, and the value used is $T_2 = 0.5$. This threshold value is chosen as it is expected that when the blob covers a significant part of the image, a reanalysis is needed as the chosen colour may no longer be unique or salient. This threshold selection is similar to the selection of T_1 ; that is, a larger T_2 will result in a potential wrong tracking while a smaller T_2 will result in unnecessary invoking of the ACF. We found that the T_2 can be chosen in the range from 0.45 to 0.55 without tracking errors, therefore a round value of 0.5 was chosen.

(3) When the speed of the blob between frames exceeds a certain threshold (sudden jump) as

$$\sqrt{\frac{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2}{X_d^2 + Y_d^2}} > T_3, \quad (4.4)$$

where X_i and Y_i are the x and y coordinates of the centroid of the blob in the i th frame, X_d and Y_d are the same as in (4.3), T_3 is a threshold, and the value used is $T_3 = 0.2$. This threshold number should vary depend on the moving velocity of the robot—as the robot in the experiment was moving in relative slow speed (~ 0.1 metre/s), and the frame rate is around 15 frames per second—this 20% movement limit of the image size is chosen. If the robot speed is increased then the value of T_3 should be increased proportionally. However, if the frame rate increases, this threshold should be decreased proportionally.

Once the ACF is invoked, the ROI is redefined according to the size of the moving object from the previous image frame. To accommodate the change in speed and position of the target object between the current and the previous image frame, the ROI's width and height for the moving object are evenly increased by N if it covers less than half of the total area of the frame, otherwise its width and height are evenly decrease by the same percentage, N . The value used is $N = 50\%$. The image is then reanalysed with the new ROI to find a new colour segment to apply. The threshold values used are based on the robot speed and frame rate and are obtained empirically.

4.3. Reducing Unnecessary Adaptation

During tracking, the colour features of the entire image are constantly updated. The largest blob that contained the selected colour feature is deemed to be the moving object. However, due to possible changes in the surrounding environment, there may be large objects in the background that has similar colour to the tracked object. To avoid the unnecessary use of the ACF (particularly the wrongly use of condition 3, (4.4), in Section 4.2), a method of “chopping” background into smaller regions has been introduced.

During the “chopping” process, a number of division lines with black pixels are introduced to cut background objects into smaller pieces. This will cut possible background objects that have similar colour to the moving object into smaller white blobs. This process is

intended to reduce the size of the blobs in the selected colour segments outside the ROI. The division lines are either horizontal or vertical, and the locations of these lines are

$$\begin{aligned}
 X_{k,\text{left}} &= \frac{2x_{\min} - x_{\max}}{k}, \\
 X_{k,\text{right}} &= X_d - \frac{X_d - 2x_{\max} + x_{\min}}{k}, \\
 Y_{k,\text{upper}} &= \frac{2y_{\min} - y_{\max}}{k}, \\
 Y_{k,\text{lower}} &= Y_d - \frac{Y_d - 2y_{\max} + y_{\min}}{k},
 \end{aligned} \tag{4.5}$$

where x_{\min} , x_{\max} , y_{\min} , y_{\max} , X_d , and Y_d are the same as in (4.3), while $X_{k,\text{left}}$, $X_{k,\text{right}}$ are the locations of the left and right vertical “chopping” lines of the k th cut ($k = 1, \dots, K$), $Y_{k,\text{upper}}$, $Y_{k,\text{lower}}$ are the upper and lower “chopping” lines of the k th cut. The K value is used to determine the number of “cuts” to the larger objects outside the ROI; in this paper $K = 8$.

As division lines (black pixels) cut through objects that share the same colour (white pixels) outside the ROI, this results in the segmentation of potential background objects in same colours (noises) into smaller sizes and avoids the possibility of unnecessarily invoking the ACF process. Since the cutting is more concentrated as it is further away from the ROI, this decreases the impact of potential noises when they are further away to the centroid of the tracked object.

To avoid “chopping” the tracked object, the ROI needs to be redefined for analysis so that the change in speed and position of the target object between the current and the previous image frame can be accommodated for the cutting process. The ROI’s width and height for the moving object are evenly increased or decreased by a certain percentage (similar to that mentioned in Section 4.2).

5. Experimental Results

A Pioneer 3 mobile robot is used to demonstrate the effectiveness of the proposed real-time adaptive colour feature identification and tracking method. A laptop with a 2.53 GHz duo Pentium 4 processor and a low cost Universal Serial Bus (USB) web camera with a maximum of 30 fps and 240×320 resolution are used to control the robot platform. The experiment is performed in the robotics laboratory that can be regarded as a crowded and noisy environment with objects of different shapes and colours presented, shown in Figure 3(a). The Pioneer 3 mobile robot is shown in Figure 3(b).

The robot is programmed using Microsoft Visual Studio C++ with Open CV, and the control of the robot is performed using the ARIA library. Although the Pioneer 3 mobile robot has many sensors, only the vision sensor provided by the web camera is used in this experiment. The robot is used to follow a randomly moving human object. The control concept of following the target object (human) is based on the area and centroid coordinates of the detected blob, and the commands and their trigger conditions are listed in Table 2. Note that the origin of the image x - y plane lies at the top left corner of the image frame as set by the camera. The movement speed of the robot and the threshold values form the trigger conditions are obtained empirically to accommodate experimental needs.

Table 2: List of commands and their trigger conditions used by the pioneer 3 mobile robot.

Commands	Trigger conditions
Drive forward at 100 mm/sec	Area falls below 4% of the entire screen
Drive backward at 100 mm/sec	Area reaches above 20% of the entire screen
Turn left 4°	Centroid lies between 21% and 40% of the maximum width
Turn left 12°	Centroid lies below 20% of the maximum width
Turn right 4°	Centroid lies between 60% and 79% of the maximum width
Turn right 12°	Centroid lies above 80% of the maximum width



(a) The robotics laboratory



(b) The Pioneer 3 mobile robot

Figure 3: The robotics laboratory and the Pioneer 3 mobile robot.

Since the focus of this research is on the adaptive colour feature identification, during experiments it is assumed that the tracked person is not to walk in narrow aisles. Therefore, the robot has no obstacle avoidance method implemented. The video frames are recorded during the operation using the USB web camera attached to the laptop computer with a rate of 10 fps during the adaptive colour filtering stage and a rate of 15 fps during the moving object tracking stage with the selected colour segment. Two different cases are considered in the experiments: simple colour (Figure 4(a)) and complex colour (Figure 4(b)) situations. In the simple colour case, the colour saliency of the target remains strong throughout the tracking, represented by the azure jumper shown in Figure 4(a). While in the complex colour case, the colour saliency of the target changes as colour of background interferes with the colour of the target continually, represented by the red jumper and black pants shown in Figure 4(b).

Key video frames of the tracking process for both cases are shown in Figures 5–8, in which the simple colour case is shown in Figures 5 and 6, while the complex colour case is shown in Figures 7 and 8. In these figures, cross represents the centroid position of the moving object, lines linked to the centroid show the paths of the moving target, the lengths of these lines indicate the moving speed of the target, the colour rendered area indicates the ROI of the moving object, and division lines for the “chopping” process are presented by the bounding boxes.

In the simple colour situation, the initial setting of the environment is shown in Figure 5(a), the background generation using IABMM is completed, and the robot is stationary. As a person walks into the scene, shown in Figure 5(b), object tracking using

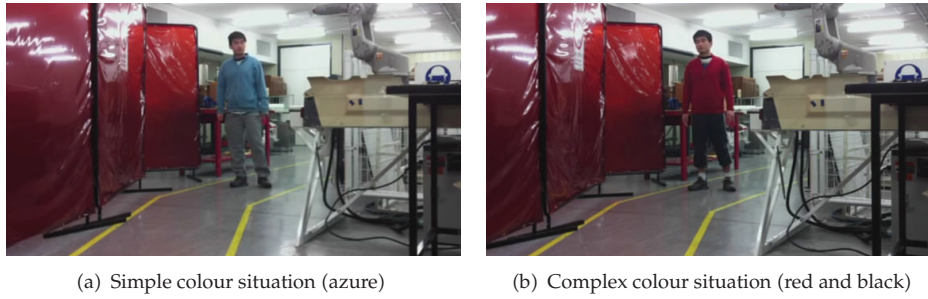


Figure 4: Two different cases used in the experiments.

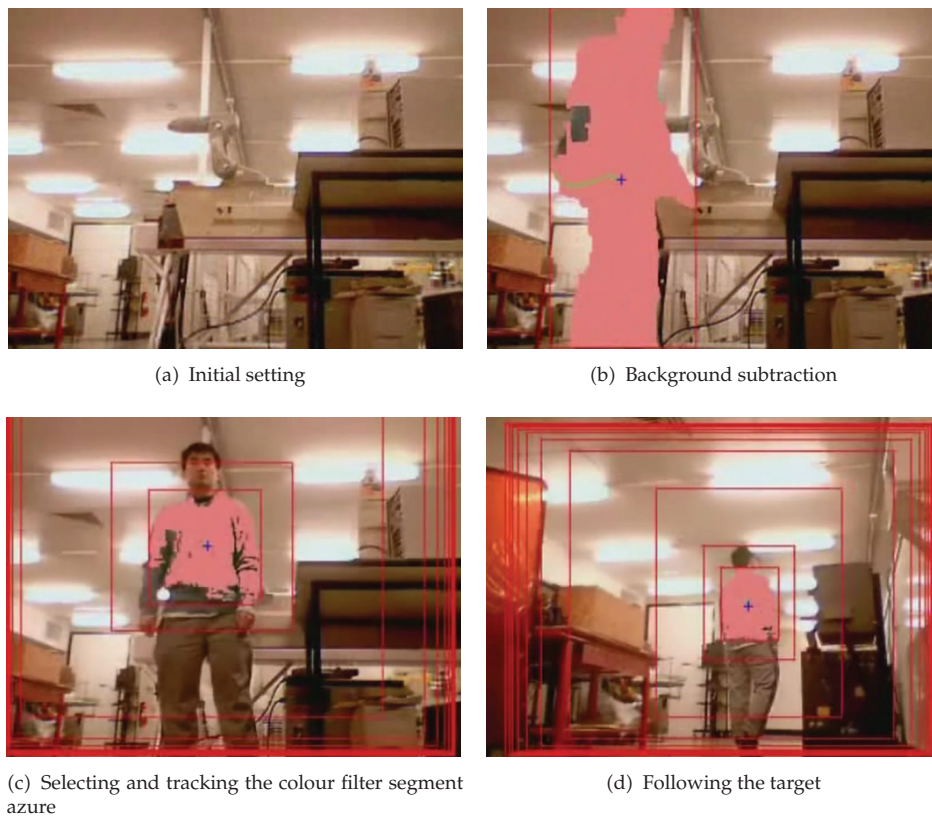


Figure 5: Key video frames of moving object tracking in the simple colour situation part 1.

the background subtraction starts to track the person. By using the ROI identified from the background subtraction, the image is filtered with each of the 15 colour filter segments, and “azure” has the highest uniqueness value U . Therefore, it is chosen to be the most suitable colour feature for tracking shown in Figure 5(c). This is because the azure jumper of the person is the most salient colour feature. The robot then tracks and follows the target person as he moves randomly around in the laboratory, shown in Figures 5(d), 6(a), and 6(b). It can be seen that the robot is adjusting the distance between itself and the tracked person,

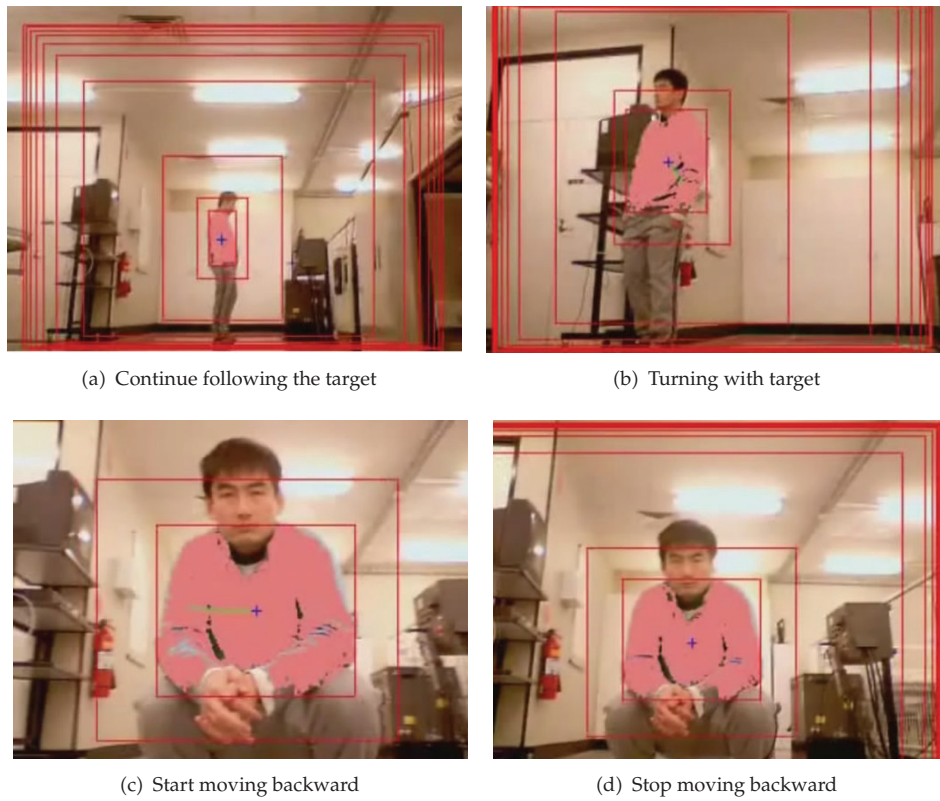


Figure 6: Key video frames of moving object tracking in the simple colour situation part 2.

shown in Figures 6(c) and 6(d). During the tracking process, there is no reanalysis using the colour adaptation method introduced in Section 4. This is because the level of interference is relatively low in the azure colour segment as the colour saliency of the target remains high throughout the experiment.

In the complex colour situation, the initial setting of the environment is shown in Figure 7(a). As a person walks into the scene, by using the ROI identified from the background subtraction, the image is filtered and “red” is chosen to be the most suitable colour feature for tracking, as the red jumper of the person is the most salient colour feature in Figure 7(b). The robot then tracks and follows the target person as he moves around in the laboratory. As the robot follows the subject and drives into a background that shares the same colour, shown in Figure 7(c), a sudden jump in number of pixels sharing the same colour is detected, and the robot starts to reanalyse due to the adaptation method of the adaptive colour filter. This result is shown in Figure 7(d). The colour filter is updated and “black” is chosen to be the most salient colour segment for tracking. Thus, the black pants of the subject are tracked, shown in Figure 8(a). When the target person moves further down the laboratory, background interference occurs due to the television and its shelf that share the same colour, shown in Figure 8(b). Thus, another jump in number of pixels sharing the same colour occurred, shown in Figure 8(c). Due to the adaptation method introduced in (4.4), the robot starts to reanalyse. The colour filter is updated and “red” is once again chosen to be the most suitable colour feature for tracking, shown in Figure 8(d).



Figure 7: Key video frames of moving object tracking in the complex colour situation part 1.

These two cases (simple colour and complex colour cases) demonstrate that the proposed method can robustly and adaptively identify a salient colour in a moving object to track. It also shows, in the second case, that the proposed thresholds can adequately cope with the changing environment and keep tracking the correct object. In comparison with existing colour-based tracking methods, the advantage of the proposed method is its capability in identifying the most salient colour feature of the moving object with a moving camera mounted on a mobile robot. This method has the ability to adapt when the identified colour lost its saliency due to changing environment. However, although not common, there are situations where colour saliency is hard to identify, this can be the results of background and foreground objects sharing one uniform colour, such as a person wearing black clothing walking in a dark room. A possible approach to solve this is by introducing a secondary feature such as shape or corner.

6. Conclusions

In this research, a new adaptive colour feature identification method for real-time moving object tracking is presented. This adaptive colour tracking method is achieved by automatically determining a unique colour of the target object by comparing to the background and updating it when required. A method of determining the region of interest of the moving

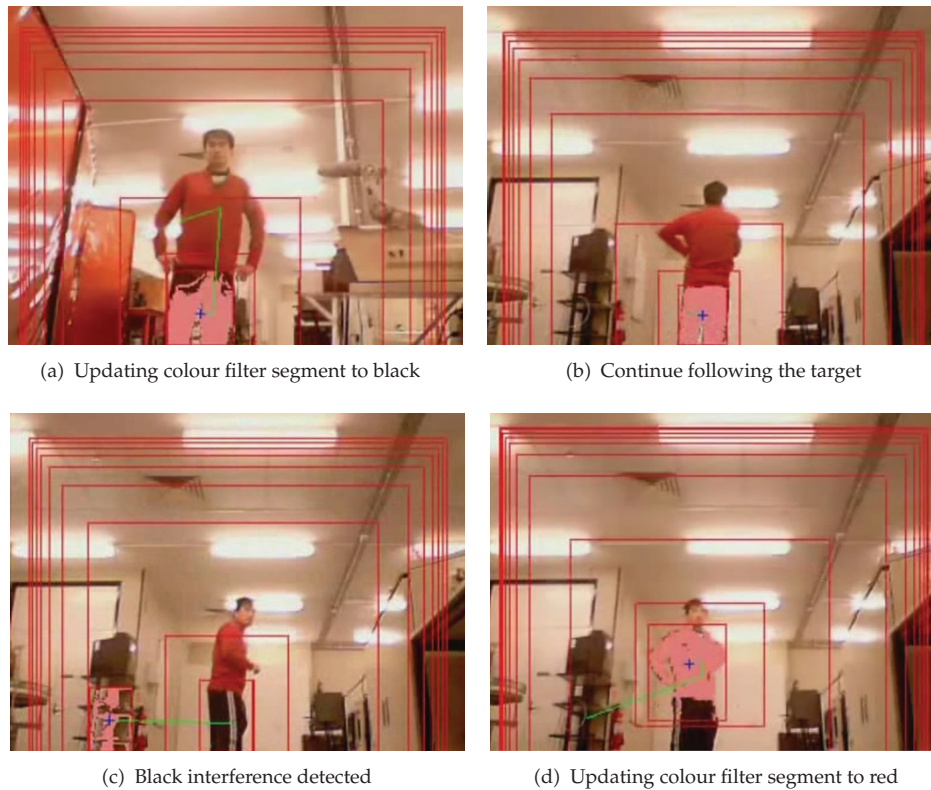


Figure 8: Key video frames of moving object tracking in the complex colour situation part 2.

target is also introduced for the adaptive colour filter to extract objects of unique colour. Experimental results show that the methods developed are reliable of motion detection and moving object tracking in a busy/crowded indoor environment.

Acknowledgment

This work is partially supported by the Australian Research Council under project ID LP099-1108 and the Lincoln Electric Company Australia.

References

- [1] V. Jupp, P. Davies, and P. Francis, *Doing Criminological Research*, SAGE, Thousand Oaks, Calif, USA, 1st edition, 2007.
- [2] C. J. Cohen, K. A. Scott, M. J. Huber, S. C. Rowe, and F. Morelli, "Behavior recognition architecture for surveillance applications," in *Proceedings of the 37th IEEE Applied Imagery Pattern Recognition Workshop (AIPR '08)*, pp. 1–8, Washinton, DC, USA, October 2008.
- [3] B. Liu, O. Jesorsky, and R. Kompe, "Robust real-time multiple object tracking in traffic scenes using an optical matrix range sensor," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 742–747, Seattle, Wash, USA, October 2007.
- [4] J. Zhu, Y. W. Lao, and Y. F. Zheng, "Object tracking in structured environments for video surveillance applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 2, pp. 223–235, 2010.

- [5] X. G. Guo, C. H. Wang, and Z. S. Qu, "Object tracking for autonomous mobile robot based on feedback of monocular-vision," in *Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications (ICIEA '07)*, pp. 467–470, Harbin, China, May 2007.
- [6] S. Y. Chen, J. W. Zhang, H. X. Zhang, N. M. Kwok, and Y. F. Li, "Intelligent lighting control for vision-based robotic manipulation," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3254–3263, 2012.
- [7] S. Y. Chen, Y. F. Li, and N. M. Kwok, "Active vision in robotic systems: a survey of recent developments," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.
- [8] S. Y. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.
- [9] S. Y. Chen, "Active vision for robotic manipulation," *Industrial Robot*, vol. 39, no. 2, pp. 111–112, 2012.
- [10] S. Y. Chen, H. Tong, and C. Cattani, "Markov models for image labeling," *Mathematical Problems in Engineering*, vol. 2012, Article ID 814356, 18 pages, 2012.
- [11] R. Yumiba, M. Miyoshi, and H. Fujiyoshi, "Moving object detection with background model based on spatio-temporal texture," in *Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV '11)*, pp. 352–359, Kona, Hawaii, USA, January 2011.
- [12] Z. Liu, K. Q. Huang, and T. N. Tian, "Cast shadow removal in a hierarchical manner using MRF," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 1, pp. 55–56, 2012.
- [13] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '99)*, vol. 2, pp. 246–252, Fort Collins, Colo, USA, June 1999.
- [14] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, vol. 2, pp. 28–31, Cambridge, UK, August 2004.
- [15] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proceedings of the European Workshop on Advanced Video Based Surveillance Systems*, pp. 1–5, London, UK, September 2001.
- [16] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.
- [17] C. H. Chen, C. Cheng, D. Page, A. Koschan, and M. Abidi, "A moving object tracked by a mobile robot with real-time obstacles avoidance capacity," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, pp. 1091–1094, Hong Kong, China, August 2006.
- [18] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.
- [19] J. Han, G. Awad, and A. Sutherland, "Automatic skin segmentation and tracking in sign language recognition," *IET Computer Vision*, vol. 3, no. 1, pp. 24–35, 2009.
- [20] Y. Wu and T. S. Huang, "Color tracking by transductive learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 133–138, Hilton Head, SC, USA, June 2000.
- [21] T. Goncalves and A. I. Comport, "Real-time direct tracking of color images in the presence of illumination variation," in *Proceedings of the IEEE International Conference on Automation Science and Engineering*, pp. 4417–4422, Shanghai, China, May, 2011.
- [22] J. Orwell, P. Remagnino, and G. A. Jones, "Multi-camera colour tracking," in *Proceedings of the IEEE Workshop on Visual Surveillance*, pp. 14–21, Fort Collins, Colo, USA, July 1999.
- [23] I. Leichter, M. Lindenbaum, and E. Rivlin, "Tracking by affine kernel transformations using color and boundary cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 164–171, 2009.
- [24] P. Li, "An adaptive binning color model for mean shift tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1293–1299, 2008.
- [25] J. Wang and Y. Yagi, "Integrating color and shape-texture features for adaptive real-time object tracking," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 235–240, 2008.
- [26] S.-H. Lee and M. G. Kang, "Motion tracking based on area and level set weighted centroid shifting," *IET Computer Vision*, vol. 4, no. 2, pp. 73–84, 2010.
- [27] T. Ojala, M. Pietikäinen, and T. Maenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [28] T. Randen and J. H. Husøy, "Filtering for texture classification: a comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291–310, 1999.

- [29] P. Gorur and B. Amrutur, "Speeded up Gaussian mixture model algorithm for background subtraction," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 386–391, Klagenfurt, Austria, August 2011.
- [30] T. S. Huang, G. J. Yang, and G. Y. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Trans Acoust Speech Signal Process*, vol. 27, no. 1, pp. 13–18, 1979.
- [31] H. Sugano and R. Miyamoto, "Parallel implementation of morphological processing on cell/BE with OpenCV interface," in *Proceedings of the 3rd International Symposium on Communications, Control, and Signal Processing (ISCCSP '08)*, pp. 578–583, St. Julian's, Malta, March 2008.
- [32] F. Chang, C. J. Chen, and C. J. Lu, "A linear-time component-labeling algorithm using contour tracing technique," *Computer Vision and Image Understanding*, vol. 93, no. 2, pp. 206–220, 2004.
- [33] H. D. Cheng, X. H. Jiang, Y. Sun, and J. Wang, "Color image segmentation: advances and prospects," *Pattern Recognition*, vol. 34, no. 12, pp. 2259–2281, 2001.
- [34] G. Fang and N. M. Kwok, "Image segmentation using adaptively selected color space," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '09)*, pp. 1838–1843, Guilin, China, December 2009.
- [35] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," in *Proceedings of the International Conference on Image Processing (ICIP '02)*, pp. 589–592, New York, NY, USA, September 2002.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

