Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE 2018
August 26-29, 2018, Quebec City, Quebec, Canada

DETC2018-85626

IMPROVED ALIGNMENT ESTIMATION FOR AUTONOMOUS DOCKING OF MOBILE ROBOTS

Shubhdildeep S. Sohal
Robotics and Mechatronics Lab
Mechanical Engineering Dept.
Virginia Tech
Blacksburg, VA, USA
shubh94@vt.edu

Wael Saab
Robotics and Mechatronics Lab
Mechanical Engineering Dept.
Virginia Tech
Blacksburg, VA, USA
waelsaab@vt.edu

Pinhas Ben-Tzvi
Robotics and Mechatronics Lab
Mechanical Engineering Dept.
Virginia Tech
Blacksburg, VA, USA
bentzvi@vt.edu

ABSTRACT

This paper presents the methodology for an improved visual tracking, intended for the autonomous control of mobile robots. We propose the cumulated use of the image processing method and positioning sensors for an improved visual tracking. The method primarily uses detection, tracking, and recognition techniques to locate the targets. The use of such methods will enable the robot to monitor the real-time continuous changes in the orientation and the alignment with respect to the target module. The proposed methodology is implemented on a subsisting genderless coupling mechanism which is integrated into a multi-directional hybrid locomotion module to test the alignment accuracy in an autonomous docking procedure. The long-term objective is to demonstrate the autonomous docking and self-reconfiguration of multiple mobile robots.

Keyword: Autonomous Docking, Image processing, Self-reconfigurable robots, Visual tracking

1 INTRODUCTION

Conventional rigid structured robots [1] are designed to perform a specific task in a repeated manner. It is easier to operate the robots under ideally controlled conditions [2], [3], but when the conditions are unknown, several additional factors come into the play. In such cases, the need arises for metamorphic robots [4-6], [7-9], which can adapt to unknown tasks and environments. The robots may combine to perform obstacle climbing locomotion [7], [10] to a surveillance [6], [11] based operation. Such robots can prove useful in environments which are either too dangerous or inaccessible to humans. These operational requirements have led to the study of reconfigurable mobile robots [7], [9].

Self-reconfigurability [8] is one such feature which enables mobile robots to align, dock and reconfigure with discrete modules to scale functionalities, thus adding an additional degree of freedom. The means of coupling provide a robust way to create a long-chained [7-9], swarm of modules. Such properties primarily relate to a rigid, and robust docking interface [12], [5]. These properties make robots more versatile and cost economical over conventional rigid structured robots and make the robot potentially capable of extreme applications. Without this interaction, robots are left with mere sensorial perception restraining their ability to act upon the surroundings.

In this paper, we discuss the challenges associated with detection and tracking of the target for a properly aligned docking. The extraction and use of visual data is an important task while designing reconfigurable robots. To address these challenges, we propose a sensing methodology based on the works of Shi-Tomasi [13] and Lucas-Kanade [14]. The method is implemented on a multi-directional mobile robot [11] equipped with a genderless coupling mechanism [12] for an autonomous reconfiguration as shown in Fig. 1.

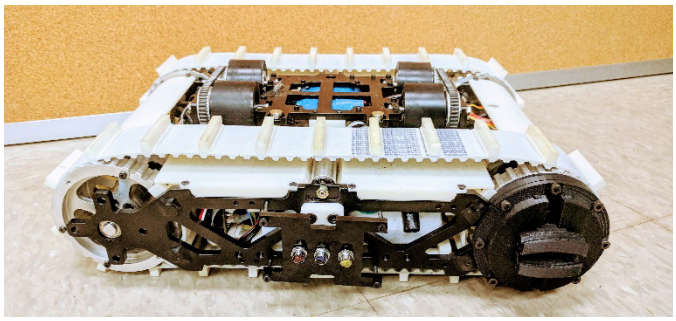

Figure 1. Hybrid Wheel-track mobile robot [11] equipped with GHEFT [12], and visual landmarks for tracking

The targets are processed using Template Matching [15], Optical Flow [13-14], [16-19], and Color Detection [20] techniques. The long-term goal of this research is to demonstrate the autonomous docking and self-reconfigurability of multiple robots to scale docking capabilities.

The next section classifies the use of sensors in reconfigurable robots. Section 3 discusses the design integration of the Visual Landmarks. Section 4 describes the sensing methodology used for the detection of a target. Section 5 discusses the experimental validation of the algorithm as integrated on the robot. It also gives a brief about the performance comparison between the 1-point and 4-point tracker. The last section concludes the experimental results and discusses the future work of our research.

2 RELATED WORK

This section covers several highly capable, state-of-the-art modular self-reconfigurable robots capable of autonomous docking. This section covers the background and provides the motivation behind the proposed methodology.

2.1 Background

The robots whose morphological properties can provide different configurations are of keen interest for research.

Over the past decade, a considerable number of studies have demonstrated the autonomous control of modular as well as mobile robots (shown in Fig. 2). S. Murata's M-TRAN [10], [21], uses a blinking pattern of LED for detection using a pinhole camera to form a long chain of swarm bots. A. Castano's CONRO [8], M. Yim's PolyBot G2 [9], G. Qiao's Transmote [22], etc. use IR sensor for a close range proximity of targets. Ji-an Xu's Tanbot [6] uses a gyro for orientation and digital camera to implement the autonomous docking. These robots have made extensive use of positioning and tilt sensors to minimize the positional and angular misalignment errors. Each module of modular robots has multiple degrees of freedom that can align with other robots with the use of sensors [10], [21-24]. These robots can maneuver and accordingly, interact with the environment.

Table 1. Classification of Sensors Used in Robots for Autonomous Control

Robot	Sensor	Coupling
CKBot[2][3]	Smart camera, IR	Magnetic linkage
Tanbot[6]	Camera, IR, Gyro	Pin and hole-latch
M-TRAN[10][21]	IR, camera	Hooks
SAMBOT[26]	IR	Hooks
MBLOCKS[23]	IR, Hall effect	Magnetic linkage
Telecubes[24]	IR, Magnetic	Magnetic linkage
Transmote[22]	IR, Angle/Tilt	Lock and key
JL II[7]	IR, GPS, Ultrasonic	Hooks-gripper
Polybot G2[9]	IR, Hall effect	Pin and hole-latch
CONRO[8]	IR	Pin and hole-latch
Swarm-bots[25]	Camera, IR	Hooks-gripper

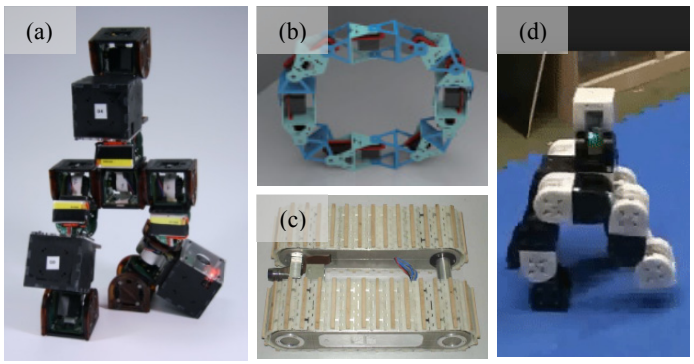

Figure 2. Self-reconfigurable robots, (a) CK-Bot [2], (b) Transmote [22], (c) Tanbot [6], (d) MTRAN [10]

As shown in Table 1, most of the modular robots rely on the use of proximity sensors, such as IR and Photodiodes, to determine the robots available in their surroundings, but their operation is limited to the sensing range. The lack of vision [7-8], [22-24] to recognize the target also pose a shortcoming for the proposed autonomy. Moreover, the autonomous docking is more difficult for mobile-type robots [4] as compared to modular robots [2], [6], [10], [25].

There arises a need to use the visual tracking to determine a long-range position and orientation navigational accuracy. The use of image processing techniques allows us to detect and track the target with certainty in the given frame of reference. These applications can be found in mobile [6], underwater [27-28] and aerial robots [29]. The combined use of visual feedback along with the positional sensors serves a better advantage over conventional positional sensors.

There are several image processing methods available to detect and track the targets effectively but each method varies based on their operational speed and accuracy. The basic requirements for a real-time detection are:

1. The method should operate in real-time conditions,
2. It should be fast, and
3. It should be robust to determine the targets with certainty even in case of a background noise.

The methods such as Color Detection [20], Template Matching [15], Harris Corner detector [30], Optical Flow [13-14], [16-19], SIFT [31], etc. are among the several methods which can be used for object recognition; however, only a few of those could be implemented for faster tracking of objects. Furthermore, due to the small size of robots, image processing is difficult to implement using onboard control. Therefore, in many cases, these methods are performed using a host PC, which provides the feedback for the autonomous control through a microcontroller.

2.2 Motivation

A method such as Color Detection has been practically demonstrated in [27], [28] for the autonomous docking of AUV. Since the detection, in this case, is only dependent on Color

Detection [20] using RGB, it is difficult to implement if the same colored objects (i.e. a different object with the same color) are present within that frame of reference [27].

1. The use of Template Matching in [28], is an effective way to overcome such obstructions, however, such methods are difficult to implement in real-time due to their performance limitations (refer to Table 4).
2. In [29], Krukowski and Perkins presented how two methods, Template Matching, and Optical Flow, can be combined together to detect the UAV even in case of a noisy background.

The use of such combination for object detection provided us with the motivation to use the Color Detection [20] method, which, combined with Template matching [15] and Optical Flow [14], can track the target effectively irrespective of any object and color obstruction.

The objective of the proposed methodology is to bring two distant robots closer to each other using path planning method and after they come in close proximity, the autonomous docking procedure is initiated. The detection process makes use of the Template Matching [15], Optical flow method [14] and Color segmentation [20] technique along with continuous rotation high-speed servo with positional feedback to adjust the position of the robot relative to the target. So far, in this paper, only the algorithm and sensing method will be discussed. Since the method is based on LED pattern detection, several different patterns can be assigned to different robots for multiple docking procedures.

3 DESIGN INTEGRATION

The related visual target, Fig. 3, for tracking the mobile robots is presented in this section to check the validity of the proposed methodology. The Hybrid Tracked-Wheeled Multi-Directional Mobile Robot [11] is used as a primary source for analyzing the positional and angular alignment data.

Visual data from sensors provide a reliable and robust way to recognize and track visual landmarks. Coupling is an important characteristic of a robot, which makes it possible for two or more robots to interact and reconfigure with each other [1], [4-6], [12].

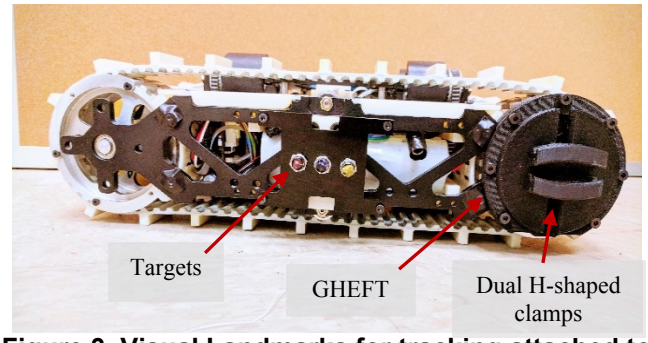

Figure 3. Visual Landmarks for tracking attached to the side frame of the Hybrid mobile robot

Genderless, High strength, Efficient, Fail-safe, and high misalignment Tolerant (GHEFT) [12] is one of the few docking mechanisms, which is flexible, can tolerate high misalignment and gives a genderless advantage over gendered coupling [6-7], [9]. The dual profile of the H-shaped clamps allows for the docking to be performed from either inside or from outside making the mechanism genderless and fail-safe when either of the mechanisms fails to actuate.

4 METHODOLOGY

In presenting the output of an imaging sensor to a human observer, it is essential to consider how the image is transformed into information by the viewer.

4.1 General Approach

Visual Image data itself represents the spatial distribution of physical quantities such as luminance and spatial frequencies of an object. The perceived information may be represented by attributes such as brightness, color, and edges. Such features are extremely important for the detection of an image as shown in [30-31]. However, a real-time detection and tracking of the target itself is a challenging task as the small delay in tracking could lead to increase in alignment error.

4.1.1 Template Matching method

Template matching is used to detect a particular section of the image within a larger image. The presence of a known object in a scene can be detected by searching for the location of the match between the object template, $u(m,n)$ and the scene, $I(m,n)$. Template matching can be conducted by searching the displacement of the object template, $u(m,n)$. Therefore, it makes the use of a template for the detection of that region of interest (template) in a particular frame of reference, such that the best match practice is given by comparing the template matrix with the image matrix,

$$\mathbf{R}(p,q) = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}[\mathbf{I}(m,n) - \mathbf{u}(m-p,n-q)]^2 \qquad (1)$$

where, $R(p,q)$ is the resulting matrix of the comparison. Also, the area correlation between the image and the template can be maximized by minimizing the mean square error of (1) as,

$$\mathbf{r}(m,n) = \mathbf{I}(m,n) * \mathbf{u}(-m,-n) \qquad (2)$$

which implies that the area correlation is equal to the convolution of the image, $I(m,n)$, and the impulse response, $u(-m,-n)$.

The method is effective when a small portion of the image, called the template, is an exact cut from a larger image in which the template has to be found; however, the method fails in case of noise interference. Also, the method is robust to find a match in the image, but it fails to produce the desired results when the matching involves the changes in scale and rotation among the template and the position in the image.

4.1.2 Optical Flow method

Optical Flow method is used to measure the displacement of a pixel within consecutive frames. The calculation of these motions helps in determining the velocity of the target in the given series of frames

$$\mathbf{I}(x, y, t) = \mathbf{I}(x + \Delta x, y + \Delta y, t + \Delta t) \tag{3}$$

where Δx, Δy, and Δt, represents the shift in the position of the image with Intensity $I(x, y, t)$ by, Δx, Δy, and Δt.

The classical algorithms often used are tracking good features by Shi-Tomasi [13], tracking the flow of pixels by Lucas-Kanade [14], using global smoothness constraint by Horne-Schunck [19], based on the motion of edges by Buxton-Buxton [17], using phase information by D. J. Fleet [18] and so on. The basic principle behind optical flow method is to first detect some Shi-Tomasi [13] corner points in the frame and then use the Lucas-Kanade method [14] to determine the flow of those points. This flow can be represented using Eq. (4), which, (assuming a small shift), is the Taylor-series expansion of Eq. (3),

$$\mathbf{I}_x \mathbf{u} + \mathbf{I}_y \mathbf{v} + \mathbf{I}_t = 0 \tag{4}$$

where, I_x, I_y, are the image gradients, I_t is the time gradient, $u = \dot{x}$, and $v = \dot{y}$, are the image flow (velocity) vectors, and can be calculated using,

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \sum_i \mathbf{I}_{x_i}^2 & \sum_i \mathbf{I}_{x_i} \mathbf{I}_{y_i} \\ \sum_i \mathbf{I}_{y_i} \mathbf{I}_{x_i} & \sum_i \mathbf{I}_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i \mathbf{I}_{x_i} \mathbf{I}_{t_i} \\ -\sum_i \mathbf{I}_{y_i} \mathbf{I}_{t_i} \end{bmatrix} \tag{5}$$

where the inverse matrix is based on the Harris-corner detector [11] and determines the points which will be tracked in the consecutive frames.

The validation of Lucas-Kanade [14] approach is dependent on the following key assumptions:

1. No change in the pixel intensity of in-between frames,
2. There is a small motion between subsequent frames, and
3. The motion of a point is similar to the motion of the neighboring pixels.

To track the robot, it is important to satisfy these assumptions in order to use the optical flow approach. The overall methodology is divided into two phases:

1. Detection and Tracking of the target,
2. Detection of the LED pattern for Orientation, Alignment and Height estimation relative to the target.

4.2 PHASE 1: Detection and Tracking of the Target

The first phase, i.e., Detection and Tracking of the robot is the most important step for a successful docking, as it will extract the location of the docking module in the given frame. Therefore, an initial template, shown in Fig. 4(a), of the side frame of the target is provided to begin the search process.

The goal is to leverage the information about the target in the frame using template matching and after finding the location of the target, Lucas-Kanade Pyramid Optical Flow [14] method is initialized to track the motion of the target. Since the template, $u(m,n)$, is searched in the whole image, $I(m,n)$, individually it is a very slow process, as shown in Table 4. The same is the case with the optical flow as it is good to track only the motion of good features but not the portion of the image. However, performance validation of each method was also done during the process as shown in Table 4.

To overcome this, detection of the docking module will be limited to a set of certain motion parameters using the search area that can be reduced for template matching and optical flow method. Once the target is found in the frame, the points are stored in the result matrix using Eq. (1). Then, the matched template is defined by finding the global minimum values in the matched matrix shown in Fig. 4(b). The corner coordinates of the matched template, Point($x_{\text{matched}}, y_{\text{matched}}$), are passed to the vector, *curr_point*, using the push-back operation. To track the motion of the matched template in the successive frames, the new position of each coordinate is passed on to give the updated velocity feedback. Since there can be same or different colored LEDs in a particular frame [27], the processing method has to be robust to eliminate such interference factors. The matched template is passed on for the LED detection as a Region Of Interest (ROI). This limited sized search window helps to reduce the search area and to speed up the process. A separate but same centered (shown in Fig. 4(c)) external region of interest (ROI_{ext}) is also defined to search for the LED pattern such that the area of ROI_{ext} is more than the matched template area, ROI.

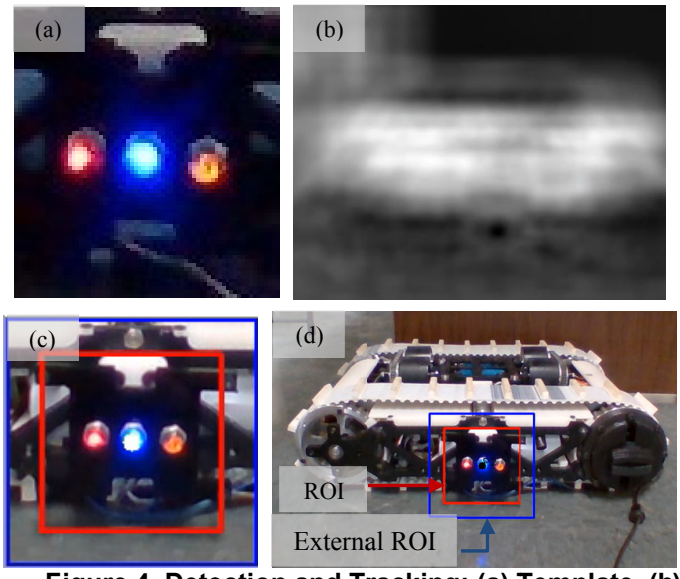

Figure 4. Detection and Tracking: (a) Template, (b) Normalized image, (c) Extracted ROI, (d) Output Image

TABLE 2. Module detection Algorithm

Module Detection and Tracking
Input: Template, target to be found **Output:** Detection and Tracking of target 1. ***Assign*** template, $u(m,n)$, for detection 2. ***Initialization,*** video capture on 3. ***Convert frame color,*** RGB to GRAY 4. ***Initiate,*** matching method=CV_TM_SQDIFF Extract corner coordinate of the matched template, Store in curr_point 5. ***while*** **(1)** 6. ***If,*** *prev_frame is empty* copy curr_frame to prev_frame **else *continue*** Initialize *LKOpticalFlow* (prev_point, curr_point) *swap*, curr_point and prev_point *swap*, curr_frame and prev_frame ***end*** (case structure@6)

The motive behind adding another feature is to eliminate all other external lights or noises which are not relevant to the docking process and furthermore, to compensate for the scaling of the image. Another advantage of this feature is to check the availability of the docking module, if available for docking or not. If the LED pattern gets detected within this search area, ROI_{ext} , then the module is available; otherwise it needs to find another available module in that frame. Therefore, the two methods, Tracking and LED pattern detection, work in a synchronized manner such that the validity of one justifies the validity of other.

In this paper, only the tracking and sensing of the target for autonomous docking has been demonstrated, but the method is also valid for multiple docking modules as each robot can be assigned with a different LED pattern.

4.3 PHASE 2: Detection of LED pattern

The second phase, LED pattern detection, involves the use of the Color Detection [20] technique for the detection of a particular LED color using HSV range (Hue, Saturation, and Value). In this method, after the target has been detected using Template Matching, the RGB frame of the ROI_{ext} is acquired (Fig. 5(b)).

The required color (to be detected) layer matrix is then extracted from the RGB frame. Before converting the RGB frame to gray image, the Median filter [32] is applied to the ROI_{ext} , to add a sense of blur, thereby removing noise from the RGB frame.

$$\mathbf{v}(m,n) = median\{\mathbf{y}(m-k,n-l),(k,l) \in \mathrm{W}\} \tag{6}$$

where $y(m,n)$ is the input image, $v(m,n)$ is the output image and W is the window size. As seen in Fig. (5), the use of HSV method makes it easier to filter colors based on the interest. The Hue of a color refers to the redness, greenness and so on, such

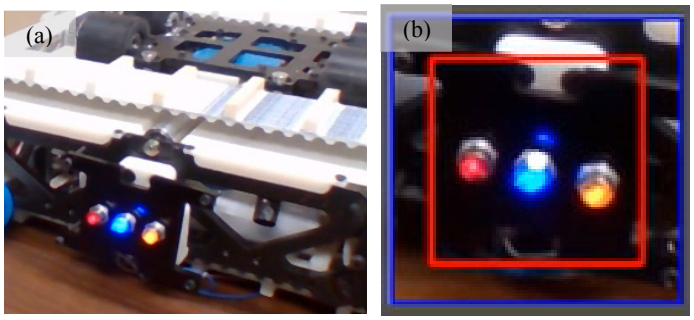

Figure 5. Thresholding of RGB image, (a) Input image, (b) Extracted ROI

that the combination of HSV gives different colors. The color conversion process from RGB to HSV is given by,

$$H' = \begin{cases} \dfrac{(G-B)}{V-\min(R,G,B)} & \text{if } V = R \\ \dfrac{(B-R)}{V-\min(R,G,B)} + 2 & \text{if } V = G \\ \dfrac{(R-G)}{V-\min(R,G,B)} + 4 & \text{if } V = B \end{cases}$$

$$H = 60^{o} \times H' \tag{7}$$

$$S = \begin{cases} \dfrac{V-\min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{if } V = 0 \end{cases}$$

$$V = \max(R,G,B)$$

Now, the RGB frame is converted into the binary image, *T(m,n)*, using the color thresholding technique. The produced binary image consists of a black pixel of 0 value and a white pixel of 255 value, such that the light source gets separated from the surroundings. As for the blue, red and green color of the LED, any color that falls beyond these value ranges gets converted into a black pixel for a given frame. It is important to note that these value ranges tend to differ with the change in light intensity in the background.

After the threshold image for the color of interest has been acquired from the RGB frame, the gray image is filtered using relevant operators such as morphological filtering [33] and Median [32]. It is possible that the gray image obtained after color thresholding contains noise in the form of white pixels. It is important to filter out these random, un-necessary pixels so as to avoid any interference with the detection performance. Therefore, *Morphological filters* [32-33] (Fig. 6) such as *Erosion* and *Dilation* [32-33] are used to remove any noise and lines or pixels while preserving spatial resolution. Erosion, $\varepsilon_B(X)$, is a shrinking operation, whereas dilation, $\delta_B(X)$, is an expansion operation. In order to remove noise by filling small holes and narrow bays in the frame of reference, dilation is used. In this process, both techniques are used twice such that erosion is accompanied by dilation.

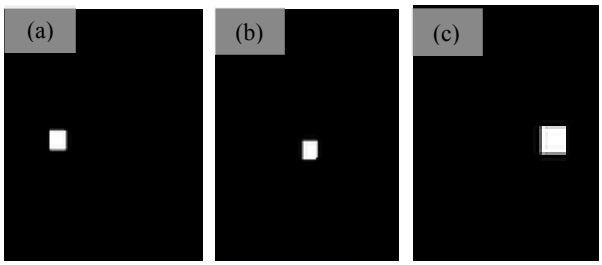

Figure 6. Applying Erosion and Dilation on threshold image: (a) for red, (b) blue, (c) yellow

For each technique, a structuring element, B, of rectangular 5 by 5 pixel was used for erosion and rectangular 7 by 7 pixel was used for dilation.

$$\begin{aligned} &\text{Erosion: } \varepsilon_B(\mathbf{T}) = \mathbf{T} \ominus \mathbf{B} \triangleq \{x : \mathbf{B}_\mathbf{x} \subset \mathbf{T}\} \\ &\text{Dilation: } \delta_B(\mathbf{T}) = \mathbf{T} \oplus \mathbf{B} \triangleq \{x : \mathbf{B}_\mathbf{x} \bigcap \mathbf{T} \neq \phi\} \end{aligned} \tag{8}$$

After the threshold image has been filtered from noise, contour finding method is implemented to define the shape of the white pixelated objects. Green's theorem [32] is used to calculate the moment for each contour, using the weighted average of the pixel intensity using

$$\mathbf{M}_{ij} = \sum_{m=1}^{N} \sum_{n=1}^{N} m^i n^j \mathbf{T}(m,n) \tag{9}$$

where i, and j are the orders of moments and $T(m,n)$ is the input threshold image.

The calculation of the moment gives the area and the centroid of each target. After these parameters have been calculated, the detected contours are filtered by defining the minimum area as 10 by 10 pixels. This minimum area sets the limit for the area of the contour such that below this value any pixelated object is considered as noise. As the targets are successfully detected, markers, as shown in Fig. 7 are used to highlight the targets in the frame of reference.

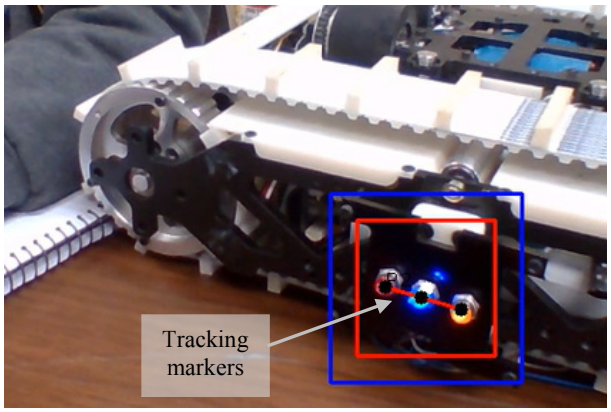

Figure 7. Output image showing tracking using LED pattern

TABLE 3. Alignment Estimation Algorithm

Robot Alignment and Height Estimation

Input: Detection and Tracking of target
Output: Angular and Height estimation of the target

7. ***Get*** *ROI* from the result matrix, R (Eq. 1)
8. ***Apply*** Median*
   Initialize thresholding, copy ROI_{ext} to threshold image
9. ***If,*** *tracked_object==true*
   Find object contour in threshold image, *X*
   Extract centroid and area of contour**
   erode *X* and ***dilate*** *X*
   else ***continue***
10. Estimate Alignment angle, θ
11. Estimate the orientation angle, $\alpha_{\text{rel}} = \alpha_{\text{target}} - \alpha_{\text{source}}$
12. Estimating the height, $\Delta h = h_{\text{target}} - h_{\text{camera}}$

end (while loop@5)

*Refer to Eq. 6, **Refer to Eq. 9

The calculated centroids represent the position of the led markers with respect to the image plane in terms of pixels as, (u_r, v_r), (u_b, v_b) and, (u_y, v_y) for red, blue, and yellow LED respectively. These corresponding pixel co-ordinates can be related to the camera's intrinsic and extrinsic parameters as shown in Eq. 10. Based on the intrinsic parameters of the camera, given by, $a = (c_u, c_v, f, \alpha)$, the projection of the 3-D point *(X, Y, Z)* in the 2-D point *(x, y)* co-ordinate system can be equated as,

$$\begin{aligned} x &= X / Z = (u - c_u) / f\alpha \\ y &= Y / Z = (v - c_v) / f \end{aligned} \tag{10}$$

where $m = (u, v)$, gives the co-ordinate of the image point expressed in the pixel units, c_u and c_v represents the co-ordinates of the principal point, f is the focal length and α is the ratio of the pixel dimensions. The representation of the pixel co-ordinates serves as the landmark for the positioning of the robot relative to the target. The depth estimation is also done using these parameters. The estimated depth is further utilized to improve the reliability of the marker, by comparing the actual length between the corner LEDs with the corresponding projected pixel length. This comparison results in a multiplication factor, which, multiplied with the height and the width of the tracker generates an adaptive marker based on the estimated depth.

5 EXPERIMENTAL VALIDATION

The Image processing is performed using Microsoft Visual Studio, C++, OpenCV libraries. A set of three distinctive LEDs is used as landmark and is attached on the side-frame of the mobile robot for experimentation [11].

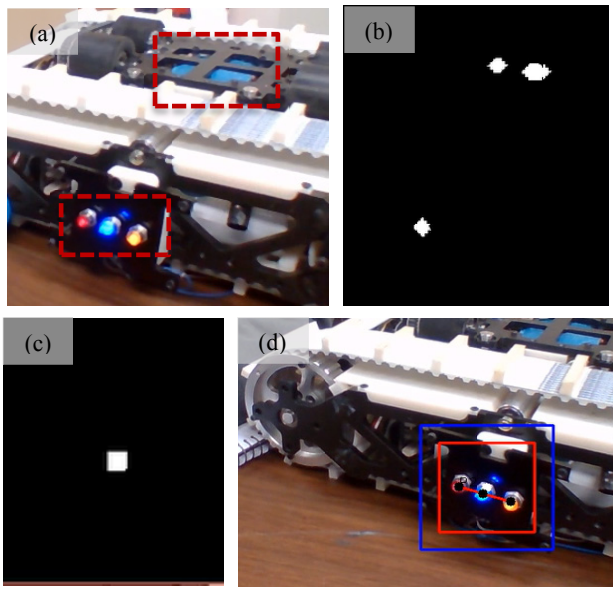

Figure 8. Detection of blue color: (a) Input, (b) Color detection without ROI, (c) with ROI, (d) Output

5.1 Experimental results

Sec. 5.1.1 compares the performance of proposed methodology with existing tracking methods. Sec. 5.1.2 explains the effect of Optical Flow point selection on the shape of the tracker. The improvement comparison in the detection of the target using the methodology and the existing color detection method is shown in Fig. 8.

5.1.1 Speed comparison test

The major concern for a real-time tracking algorithm is the tracking rate and its reliability against the noisy background. The test against the noisy background has already been discussed in the previous section, speed comparison test is performed to determine the real-time performance of different processing methods for a 640 x 480 image frame. As shown in Table 4, the Target detection rate using different search area, e.g. Entire Image (EI), and Motion Limited (ML), is observed and the corresponding frames per sec (fps) are calculated. The fps is measured by initializing a counter for every frame and recorded for a limited number of seconds. A set of three values was recorded for each method and was averaged to obtain the approximate fps.

TABLE 4. Performance Comparison of Real-Time Tracking Algorithms

Method	Search Area	No. of fps for 60 sec tracking			
		1	2	3	Avg.
TM**	EI	9.06	9.11	9.04	9.07
OF	EI	29.98	29.92	29.93	29.94
CD	EI	16.56	16.31	16.23	16.36
TM + OF*	ML	29.83	29.68	29.73	29.74
TM + OF + CD*	ML	22.54	22.26	22.34	22.38

**Methodology used in the paper, **Modified for Real-Time detection*

where TM + OF and TM + OF + CD represent the Template Matching combined with Optical Flow and Template Matching combined with Optical Flow and Color Detection method, respectively.

The Template Matching method has been modified for a real-time tracking using the conventional Template Matching method. As shown in Table 4, the combined methodology of TM and OF (fps = 29.74) is comparable to the individual OF (fps = 29.94) tracking. There is a 24.7% decrease in the performance if OF is compared with TM, with OF and CD combined. However, the frame rate is 22.38, which is 37.9% better than the individual CD method and more than twice the frame rate of the TM method (modified for real-time detection). Hence, the combined methodology of Template Matching, Optical Flow, and Color Detection is used for this research.

5.1.2 Effect of point selection on the tracker

The changing shape is an important aspect of detection when the robot is moving towards or away from the target, as the changes in the position of the camera could lead to error in tracking due to lack of scalability, as shown in Fig. 9(e). In this test, the proposed methodology is tested for sparse LK Optical Flow using $P_{center} = (x_{c0}, y_{c0})$ and $P_{corner} = \{P_{c1}, P_{c2}, P_{c3}, P_{c4}\}$, i.e. for the point at the center and points at the corners of the matched template as shown in Eq. (11). The performance of both extraction methods is observed and recorded simultaneously. The expected ideal behavior of tracker and the actual behavior are shown in Fig. 11. The benefits of a corner point tracking over a central point tracking are:

1. The independent (corner) point tracking allows for multi-scale tracking as compared to a fixed scale in central tracking (refer to Fig. 9).
2. The tracking allows for the change in the orientation of the tracker in the same way as that of the object as shown in

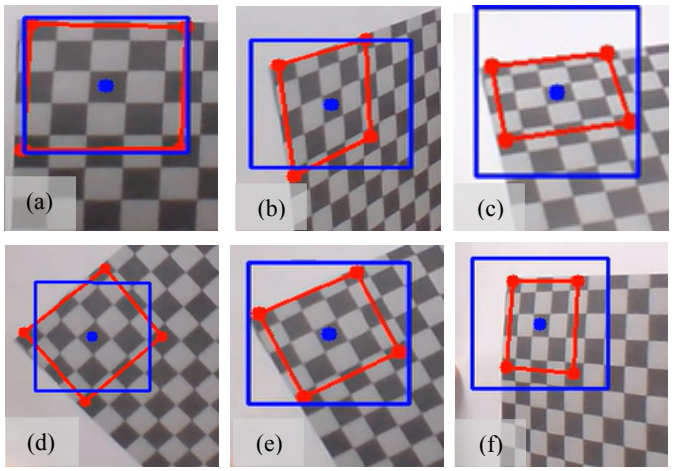

Figure 9. Comparing the actual 1-point (blue), and 4-point (red) tracker: (a) Front, (b) Tilt, (c) Tilt and Rotation, (d) Rotation, (e) Scaling, (f) Tilt

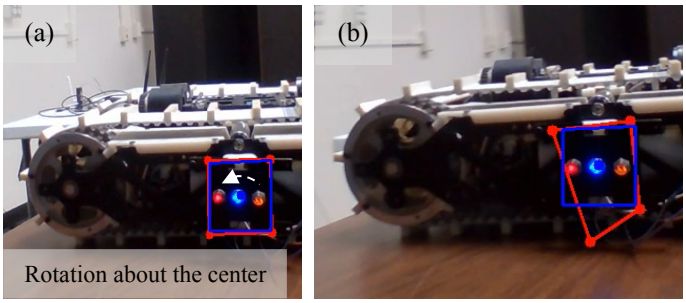

Figure 10. The inconsistent shape of the 4-point tracker compared to a 1-point tracker: (a) Initial shape, (b) shape after a full rotation

Fig. 9(d). The positional coordinates for both methods are given by,

$$\begin{aligned} P_{center} &= (x_{c0}, y_{c0}) = \left(x_m + \frac{1}{2}(n(\text{cols}))_{temp}, y_m + \frac{1}{2}(n(\text{rows}))_{temp}\right) \\ P_{c1} &= (x_{c1}, y_{c1}) = (x_m, y_m) \\ P_{c2} &= (x_{c2}, y_{c2}) = (x_m + (n(\text{cols}))_{temp}, y_m) \\ P_{c3} &= (x_{c3}, y_{c3}) = (x_m + (n(\text{cols}))_{temp}, y_m + (n(\text{rows}))_{temp}) \\ P_{c4} &= (x_{c4}, y_{c4}) = (x_m, y_m + (n(\text{rows}))_{temp}) \end{aligned} \tag{11}$$

where x_m and y_m are the coordinates of the matched template point, $(x_{matched}, y_{matched})$, and $n(\text{cols}), n(\text{rows})$ represents the no. of columns and rows of the assigned template (refer to Sec. 4.2) respectively. Fig. 9 represents two tracking methods (as shown in blue and red). The blue tracker is initialized using one central point, P_{center}, such that the rotation of the object about this point yields no change in shape or orientation of the tracker.

However, depending on the shape of the object, the shape of the tracker may vary, as shown in the Fig. 9 and Fig. 10. Therefore, for the 4-point tracker, from Fig. 10, it can be concluded that,

$$\begin{aligned} &\text{Ideal: } |v_i| - |v_j| = 0, \forall\, i, j \in \{1,2,3,4\} \\ &\text{Actual: } |v_i| - |v_j| = n, \forall\, i, j \in \{1,2,3,4\}, n \in \mathbb{R} \end{aligned} \tag{12}$$

Although both the methods are well applicable to a small range of (pixelated) motion as shown in Fig. 9, the four points optical flow is inconsistent due to independent motion parameters for each point as compared to the single point motion. As the four points behave differently in the given frame, each point has a different flow vector as shown in Fig 11. Moreover, the inter-dependence of ROI_{ext}, as mentioned in Sec 4.2, on the tracker also increases the rate of error in this case. However, the current version of the 4-point tracker is only stable for an image with edges and corners; otherwise, it fails to

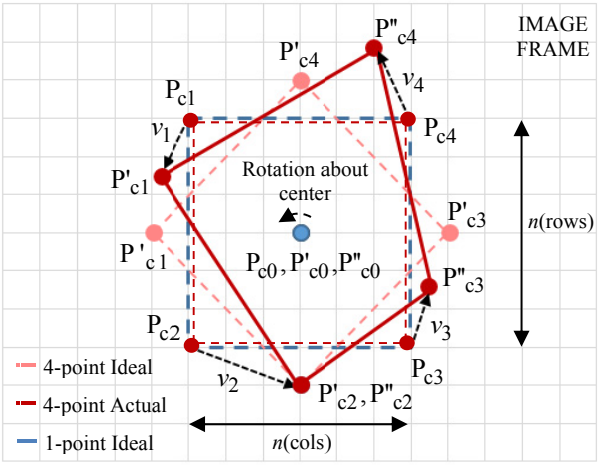

Figure 11. Actual behavior vs. Ideal behavior for 1-point and 4-point tracker

maintain consistency with the shape, as shown in Fig. 10, but further changes could be made to prevent the drifting of points.

Since the objective of this research is to track the object in real-time, 1-point OF method is used as a tracker and position estimation is done using LED targets, as the above-mentioned tracking method is much more reliable for this research.

6 CONCLUSION AND FUTURE WORK

The present work incorporates the use of image processing method which aids in the detection and alignment of two robots parallel to each other. The target tracking algorithm is used for the detection of the given template followed by the sparse optical flow algorithm. The algorithm reduces the search area by extracting the *ROI* to speed-up the process and thereby reducing the processing time. The use of *ROI* specifically increases the quality of tracking by eliminating the overwhelming majority of bad features. Such discrimination is important when the motion of the robot is autonomous. The small displacements of the robot satisfy the conditions for the LK algorithm. The results obtained were encouraging to be implemented on a reconfigurable robot capable of docking.

An area of future work involves combining the tracking algorithm with the vision control system for experimental validation of the control algorithm. It involves the use of a reconfigurable locomotion and manipulator modules to demonstrate the autonomous docking and versatility of mobile robots. The end goal is to make use of additional sensors to determine the self-reconfiguration and shape transformation based on the size of the obstacle used for scalability.

REFERENCES

[1] Ben-Tzvi, P., Goldenberg, A. A., and Zu, J. W.," Articulated hybrid mobile robot mechanism with composed mobility and manipulation and onboard wireless sensor/actuator control interfaces," Mechatronics, vol. 20, no. 6, pp. 627-639, Sep. 2010.

[2] Shirmohammadi, B., Taylor, C., Yim, M., Sastra, J., Park, M.," Using Smart Cameras to Localize Self-Assembling Modular

Robots", Proc. Of the 1st ACM/IEEE International Conference on Distributed Smart Cameras, pp. 76-80, 2007.
[3] Park, M., Chitta, S., a. Teichman, and M. Yim," Automatic Configuration Recognition Methods in Modular Robots," Int. J. Rob. Res., vol. 27, no. 3-4, pp. 403-421, 2008.
[4] Moubarak, P. M., and Ben-Tzvi, P.," On the Dual-Rod Slider Rocker Mechanism and its applications to Tristate Rigid Active Docking," Journal of Mechanisms and Robotics, vol. 5, no. 1, pp. 011010, 2013.
[5] Moubarak, P. M., Alvarez, E. J., and Ben-Tzvi, P.," Reconfiguring a Modular Robot into a Humanoid Formation: A Multi-Body Dynamic Perspective On Motion Scheduling for Modules and Their Assemblies." pp. 687-692, 2013.
[6] Zhong, M., Li, M., Sun, L.,"Tanbot: A Mobile Self-Reconfigurable Robot Enhanced with Embedded Positioning Module", Intelligent Robotics and Applications: First International Conference, ICIRA, China, 2008.
[7] Wang, W., Yu, W., Zhang, H.," JL-2: A Mobile Multi-Robot System with Docking and Manipulating Capabilities", International Journal of Advanced Robotic Systems, vol. 7, issue 1, 2010.
[8] A. Castano, A. Behar, and P. M. Will," The CONRO modules for reconfigurable robots," IEEE/ASME Trans. Mechatronics, vol. 7, no. 4, pp. 403-409, 2002.
[9] M. Yim, D. G. Duff, and K. D. Roufas," PolyBot: a modular reconfigurable robot," Proc. 2000 ICRA. Millen. Conf. IEEE Int. Conf. Robot. Autom. Symp. Proc. (Cat. No. 00CH37065), vol. 1, pp. 514-520, 2000.
[10] Murata, S., Kakomura, K., and Kurokawa, H., "Toward a scalable modular robotic System-Navigation, docking and integration of M-TRAN", IEEE Robotics & Automation Magazine, vol. 14-4, pp. 56-63, 2008.
[11] Kumar, P., Saab, W., Ben-Tzvi, P., "Design of a Multi-Directional Hybrid-Locomotion Modular Robot with Feedforward Stability Control", Proceedings of the 2017 ASME IDETC/CIE, 41st mechanisms & Robotics Conference, Cleveland, Ohio, Aug. 6-9, 2017.
[12] Saab, W., Ben-Tzvi, P.," A Genderless Coupling Mechanism with 6-DOF Misalignment Capability for Modular Self-Reconfigurable Robots", Journal of Mechanisms and Robotics, Transactions of the ASME, Vol. 8, Issue 6, pp. 061014:1-9, December 2016.
[13] Shi, J., Tomasi, C.," Good Features to Track", IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, 1994.
[14] Lucas, Bruce D., Kanade, T., "An Iterative Image Registration Technique with an Application to Stereo Vision", 7th International Joint Conference on Artificial Intelligence, pg. 674-679, 1981.
[15] Girod, B., *Template Matching 2*, EE 368 Digital Image Processing, Stanford University, 2013.
[16] Farnebäck, G., "Two-Frame Motion Estimation Based on Polynomial Expansion", 13th Scandinavian Conference, SCIA Halmstad, Sweden, 2003.
[17] Buxton, B. F., Buxton, H.," Computation of optic flow from the motion of edge features in image sequences", Image and Vision Computing, vol. 2, issue 2, pg. 59-75, 1984.
[18] Fleet, David J., Jepson, A. D.," Computation of Component Image Velocity from Local Phase Information", International Journal of Computer Vision, vol. 5, pages 77-104, 1990.
[19] Horn, Berthold K. P., Schunck, Brian G.," Determining Optical Flow", Artificial Intelligence, vol. 17, issue 1, pg. 185-203, 1981.
[20] Queiroz, R. L., Braun, K. M.," Color to Gray and Back: Color Embedding into Textured Gray Images, IEEE Trans. Image Processing, vol. 15, no. 6, pp. 1464-1470, 2006.
[21] Murata, S., Kakomura, K., and Kurokawa, H.," Docking Experiments of a modular robot by visual feedback," International Conference on Intelligent Robots and Systems, pp. 625-630, 2006.
[22] Qiao, G., Song, G., Zhang, J., Sun, H., Wang, W., Song, A.," Design of Transmote: A modular self-reconfigurable robot with versatile transformation capabilities", Proc. Of IEEE International Conference on Robotics and Biomimetics, pp. 1331-1336, Guangzhou, 2012.
[23] Romanishin, J. W., Gilpin, K., and Rus, D.,"M-blocks: Momentum-driven, magnetic modular robots", International Conference on Intelligent Robots and Systems, pp. 4288-4295, 2013.
[24] Suh, J. W., Homas, S. B., and Yim, M.," Telecubes: Mechanical Design of a Module for Self-Reconfigurable Robotics", in IEEE Intl. Conf. on Robotics and Automation(ICRA), pp. 4095-4101, 2002.
[25] Groß, R., Bonani, M., Mondada, F., Dorigo, M.," Autonomous self-assembly in a swarm-bot", Int. Symp. Autom. Minirobots Res. Edutainment, 2006: pp. 314-322. doi: 10.1007/3-540-29344-2_47.
[26] Wei, H. X., Li, H. Y., Guan, Y., Li, Y. D.," A dynamics based two-stage path model for the docking navigation of a self-assembly modular robot (Sambot), Robotica. 34 (2016) 1517-1528.
[27] Yahya, M. F., Arshad, M. R.," Development of an Autonomous Underwater Vehicle (AUV) for Shallow Water Inspection and Monitoring with Novel Tracking and Acoustic System" IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), 2015.
[28] Han, K., Lee, Y., Choi, H.," Developing an Efficient Landmark for Autonomous Docking Tasks of Underwater Robots", 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 357-361, 2012.
[29] Krukowski, S., Perkins, A.," Tracking of Small Unmanned Aerial Vehicles", Stanford University.
[30] Harris, C., Stephens, M.," A Combined Corner and Edge Detector", In Proc. Of Fourth Alvey Vision Conference, pages 147-151, 1988.
[31] Lowe, D.G., "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, vol. 60, issue 2, pp. 91-110, 2004.
[32] Jain, A. K.," Fundamentals of Digital Image Processing", Upper Saddle River: Prentice-Hall, Inc., 1989.
[33] Haralick, R. M., Sternberg, S. R., Zhuang, X.," Image Analysis Using Mathematical Morphology", IEEE Transactions on Pattern Analysis and Machine Intelligence", vol. 9, no. 4, pg. 532-550, 1987.