IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 April 2023, accepted 19 April 2023, date of publication 24 April 2023, date of current version 1 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3269796

RESEARCH ARTICLE

Segmentation-Based Angular Position Estimation Algorithm for Dynamic Path Planning by a Person-Following Robot

ISAAC ASANTE[1], (Member, IEEE), LAU BEE THENG[1], (Senior Member, IEEE), MARK TEE KIT TSUN[1], (Member, IEEE), HUDYJAYA SISWOYO JO[1], (Senior Member, IEEE), AND CHRIS MCCARTHY[2]

[1]Faculty of Engineering, Computing and Science, Swinburne University of Technology, Kuching, Sarawak 93350, Malaysia
[2]Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, VIC 3122, Australia

Corresponding author: Isaac Asante (iasante@swinburne.edu.my)

This work was supported by the Research Grant from the Malaysian Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS) under Grant FRGS/1/2020/ICT02/SWIN/03/1.

ABSTRACT This study designed, developed, and evaluated a deep-learning-based companion robot prototype for indoor navigation and obstacle avoidance using an RGB-D camera as the sole input sensor. This study proposed a dynamic path planning (DPP) method that combines instance image segmentation and elementary matrix calculations to enable a robot to identify the angular position of entities in its surroundings. The DPP method fuses visual and depth information for scene understanding and path estimation with reduced computation resources. A simulated environment assessed the robot's path-planning ability through computer vision. The DPP method enables the person-following robot to perform intelligent curve manipulation for safe path planning to avoid objects in the initial trajectory. The approach offers a unique and straightforward technique for scene understanding without the burden of extensive neural network configuration. Its modular architecture and flexibility make it a promising candidate for future development and refinement in this domain. Its effectiveness in collision prevention and path planning has potential implications for various applications, including medical robotics.

INDEX TERMS Human–robot interaction, image segmentation, object detection, path planning.

I. INTRODUCTION

Achieving effective robot-person following is a complex task requiring a deep appreciation of the challenges of developing autonomous companion robots. It is useful to segregate various subcategories, including perception, path planning, locomotion, and continuous target tracking, to address the dilemmas inherent to this field [1], [2]. Modern techniques, which rely on computer vision for environmental data input, use advanced object detection methods to obtain data on detected objects' locations in a scene. However, these detection frameworks do not provide direct information on the angular position of obstacles and target persons in images relative to a specific point or coordinate. Implementing reliable path-planning algorithms for vision-based navigation systems is also challenging, as it requires precise knowledge of object dimensions, proximity values, and optimal angles. This work focuses on processing data perceived by a person-following companion robot equipped with a depth camera. The steps involve using visual data to extract information about the robot's surroundings, including the target person and any obstacles. By processing RGB and depth information, the robot can determine the position of elements relative to itself [3] and use image processing algorithms to compute a safe path it can follow in real-time, enabling effective robot-person following.

Some previous related works are discussed in Section II. Section III presents an overview of the approach. Section IV and Section V break down the modules designed and developed, comprising comprehensive explanations and illustrations of the algorithms proposed for input image processing and dynamic path planning. The Evaluations

The associate editor coordinating the review of this manuscript and approving it for publication was Amin Zehtabian.

section (Section VI) discusses the results of experiments that test the efficacy of the proposed algorithms under various environmental conditions. This includes Angular Position Estimation and Region of Interest Extraction algorithms, and dynamic path planning modules intended to guide a companion robot during navigation toward a target person, even in cases of partial occlusion, in a simulated environment. The conclusion summarizes the main findings of the research and lists potential areas of improvement and extension for future work.

II. RELATED WORKS

Diverse options are available today for implementing robust perception in a robot navigation model. While laser and sonic technologies have proven effective, modern studies have widely adopted cameras due to their flexibility and ability to simplify algorithm development using computer vision with minimal processing power. Common choices for visual input sensors include monocular and stereo cameras [4], [5], while depth cameras are frequently employed when fusing pixel and distance information is necessary for controlling the robot's movements [6], [7], [8], [9]. To ensure safe navigation, a companion robot must remain cognizant of the target's location within the environment while avoiding collisions with other objects. Thus, distinguishing between a person and an object is of utmost importance, necessitating object detection algorithms.

In recent years, state-of-the-art machine learning models have facilitated object detection using visual data, exemplified by the TensorFlow Object Detection API [10], which provides numerous robust options including CenterNet [11], EfficientDet [12], SSD (Single Shot Detector) MobileNet [13], SSD ResNet [14], and Faster R-CNN (Region-based Convolutional Neural Network) [15]. Although these models typically output bounding boxes to indicate the position of classified objects in input images [16], they may become unsuitable for scene perception or understanding by a robot, particularly in cases where exact pixel classification is crucial. Image segmentation can be leveraged to assign labels to pixels and generate segmentation masks with significantly more accurate estimations than generic bounding boxes, which is particularly valuable for minimizing errors when focusing on targets for tasks involving object- or person-tracking.

The Mask R-CNN framework for image segmentation, developed by a group of Facebook AI (Artificial Intelligence) researchers in 2017, has been widely used in numerous studies, achieving a mask AP (Average Precision) of 35.7 when tested on the COCO dataset with ResNet-101 as the backbone [17], while running at five frames per second. One of the practical applications of this framework is video segmentation, which allows for tracking pixel masks in a video sequence and enables multi-object tracking. Researchers in [18] developed an algorithm for semi-supervised Video Object Segmentation (VOS) using Mask R-CNN to generate coarse object proposals on each video frame. This work was extended in [19] in the context of scene understanding, where a solution for Video Instance Segmentation (VIS) was designed using Mask R-CNN for detection and classification, albeit with a modified ResNeXt-101 network architecture [20].

In [21], the notion of simultaneously performing detection, segmentation, and tracking of object instances in videos is explored, which is fundamental to VIS and is formally defined, with autonomous driving mentioned as a relevant objective. The study resulted in the development of Mask-Track R-CNN, a novel algorithm based on Mask R-CNN, which was augmented via a new tracking branch to leverage the cue of appearance similarity and external memory to track object instances. IBM Research and Rutgers University [22] also conducted similar research, using Mask R-CNN as the base for a variational autoencoder (VAE) to address the Video Instance Segmentation Tracking (VIST) problem.

Over the years, researchers proposed various modifications to the Mask R-CNN architecture, such as exploring different backbone networks. These modifications have improved performance in multiple applications, such as object tracking, segmentation, and detection. Furthermore, the ability to reuse and build upon existing Mask R-CNN models can save time and resources in developing novel solutions, enabling researchers to focus on advancing the state-of-the-art in vision-based research. Overall, the flexibility and extensibility of Mask R-CNN make it a powerful tool for researchers exploring novel ideas in computer vision.

III. OVERVIEW OF THE APPROACH

While the existing methods for video instance segmentation concentrate on tracking instances across frames and provide helpful information to analyze the motion pattern of different objects, this work leans toward the more traditional image instance segmentation despite processing data from a live camera input feed.

In the context of companion robot navigation involving a camera and an instance segmentation framework, a more recent state-of-the-art model such as YOLOv8 may be preferred over Mask R-CNN due to its faster object detection speed and overall higher performance. After all, companion robots are designed to assist humans in various activities, and they require a rapid and accurate perception of the environment to ensure safety and efficiency. YOLO's single-shot detection algorithm directly predicts object bounding boxes and class probabilities from the input image, making it faster and more suitable for real-time applications. In contrast, Mask R-CNN's two-stage approach and more complex post-processing steps make it slower than YOLO. Therefore, it is justifiable to select a recent release of YOLO for faster speed and high performance.

Nevertheless, this work proposes a robot navigation solution based solely on camera input, which does not rely on the performance of the segmentation framework. Hence, Mask R-CNN is the selected model. This adaptable approach improves the system's robustness and reliability, making it more flexible for future modifications and upgrades.

This work proposes enhancing the companion robot domain with a person-following navigation model. The work introduces a novel method called Dynamic Path Planning (DPP), which utilizes innovative image processing techniques based on instance segmentation, an angular position estimation algorithm (APEA), a region of interest (ROI) extraction algorithm, a simple triangulation technique, and dynamic control points for path planning. Specifically, the depth camera mounted on the robot is used to compute the angular position of objects in a scene relative to the robot's central point of vision. A deep-learning-based object detection framework is utilized to generate instance segmentation masks of the target person and environmental objects, which are resilient to lighting changes. As shown in Fig. 1, a series of fast-running image processing algorithms is proposed to process the prediction results and extract the angular position of the target person and obstacles from the scene to control the robot's yaw and generate navigational values to aid the robot in following the target.

Depth information is integrated to continuously calculate the distance and dimension of obstacles and accurately estimate the angles required for path planning during the person-following activity by the companion robot in real time. This approach may be integrated with other techniques, such as visual-servoing-based robot navigation for modular applications [23], [24]. The values generated by the DPP technique facilitate the person-following activity by a companion robot in low-cost applications without other assistive technologies and sensors such as laser scanners, ultrasonic range finders, or wearables.

The process for estimating the target person's position and navigating the robot toward them while avoiding obstacles in the proposed companion robot prototype is outlined in this paragraph. As illustrated in Fig. 1, the input frame captured by the robot's camera is processed by the segmentation framework, which involves pixel reassignment and generates a normalized vector holding positional data of entities in the environment. Subsequently, the region of interest is computed, providing the target person's angular position and free space around them with minimal object interference. Values generated by the triangulation algorithm and a Bezier Curve-based path estimation technique are utilized to navigate toward the target person while avoiding obstacles. The resulting safe path is divided into smaller tracks the robot can follow and verify iteratively throughout its navigation and person-following task.

It is essential to clarify that the DPP approach introduced in this work does not aim to supplant current robot navigation methodologies. Instead, it proposes an affordable and dependable solution for vision-based companion robot systems. As previously mentioned, modern object detection frameworks do not provide direct information on the angular position of obstacles and target persons in images relative to a specific point or coordinate. This complicates the development of a vision-based, end-to-end robot navigation prototype. DPP addresses this problem using a multi-stage process that generates various metrics, allowing a robot to navigate securely and efficiently utilizing solely a single depth camera for environmental input data.

IV. ANGULAR POSITION ESTIMATION ALGORITHM

The prototype presented in this work employs the TensorFlow backend with Keras, Python 3, and the Mask R-CNN framework to develop the first part of the solution, which entails estimating the angular position of objects and the target person in a scene using a visual sensor. The input images are assumed to be the RGB video frames captured by the mobile robot's camera at consistent intervals. This section aims to demonstrate how two-dimensional data from an image can provide valuable information to control a mobile robot's rotation angle during navigation in an unknown environment with minimal computational cost. The aim is to provide a simulated model in the Robot Operating System (ROS), thus accurate robot rotation control can be achieved by precisely estimating the yaw value to publish to the robot's odometry topic. It is worth noting that odometry messages describing a robot's orientation in free space use quaternions in ROS [25]. Nevertheless, the Transformations library enables the attainment of Euler angles from quaternions programmatically [26].

In this work, the open-source implementation of Mask R-CNN by Matterport [27] is used for image instance segmentations. The Mask R-CNN architecture, which is built on FPN and ResNet-101, has been shown to have higher mean average precision than other convnet options such as VGG-16 in benchmark tests involving the PASCAL VOC (2007 and 2012) and COCO datasets. The selected segmentation framework repository includes pre-trained weights [28] for the Microsoft COCO dataset and can predict at least 80 classes, though with mediocre accuracy. Many of these classes correspond to objects commonly found in indoor environments. Although selecting a moderate object detection model increases the risk of poor scene understanding, the iterative DPP method proposed in this study addresses this weakness by allowing for classification errors or omissions. Thus, this method presents a generalized approach that can handle such errors and produce robust results.

A. PROCESSING SEGMENTATION MASKS

Mask R-CNN produces instance segmentation masks for each classification in a matrix of shape (h, w, n), where h and w are the height and width of the input image, respectively, and n is the count of generated masks. These individual masks must be merged into a single 2D matrix to visualize all segmentation data. However, before consolidation, each mask requires further processing to retain class information, which is crucial for the mobile robot's autonomous navigation without collision. Failure to maintain this information would make distinguishing a target person from other objects in the environment impossible. Each mask obtained from the

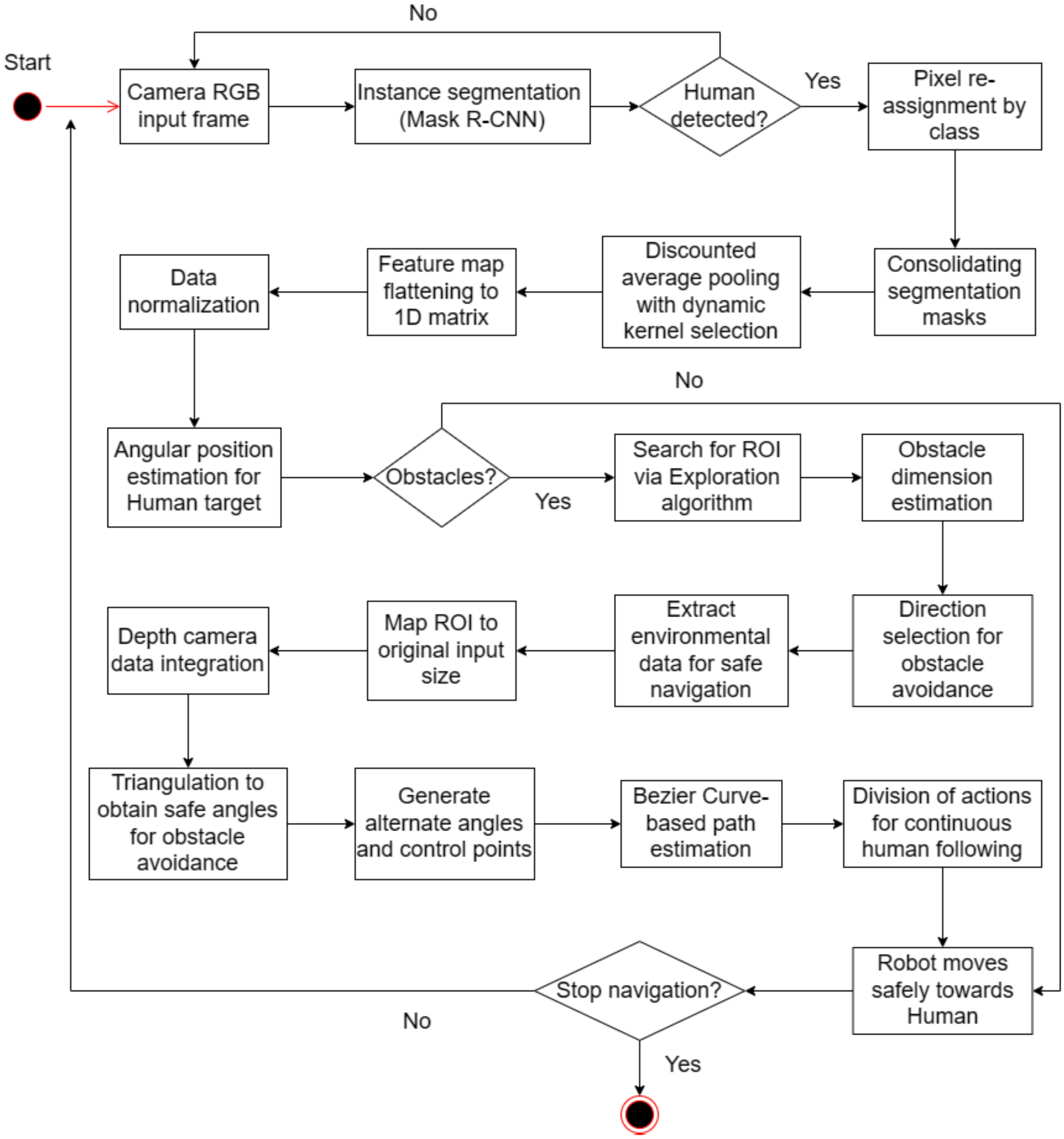

Start
Camera RGB input frame
Instance segmentation (Mask R-CNN)
Human detected?
No
Yes
Pixel re-assignment by class
Consolidating segmentation masks
Discounted average pooling with dynamic kernel selection
Feature map flattening to 1D matrix
Data normalization
Angular position estimation for Human target
Obstacles?
No
Yes
Search for ROI via Exploration algorithm
Obstacle dimension estimation
Direction selection for obstacle avoidance
Extract environmental data for safe navigation
Map ROI to original input size
Depth camera data integration
Triangulation to obtain safe angles for obstacle avoidance
Generate alternate angles and control points
Bezier Curve-based path estimation
Division of actions for continuous human following
Robot moves safely towards Human
Stop navigation?
No
Yes

FIGURE 1. Overview of the dynamic path planning method, detailing the process from the input to the robot's navigation decision.

Mask R-CNN model's predictions comprises Boolean values for every pixel in the input image.

It is crucial to differentiate between masks belonging to the 'person' class and those belonging to non-human elements in the environment; therefore, the numerical values for each mask must be updated based on the associated class. This updating of values necessitates the conversion of Boolean values to integers, with a value of 1 indicating that the pixel belongs to the corresponding mask and 0 indicating that it belongs to another mask or does not belong to any class. It is worth noting that the class names follow those from the COCO dataset used to train the Mask R-CNN model in this prototype. By retaining the class information on every frame from the camera's live feed, the mobile robot can avoid collisions while navigating autonomously, which is crucial for its effective operation.

The reassignment of pixel values for each mask returned from the prediction results follows a specific methodology that leverages a set of {-4, 0, 4}, as shown in Table 1. If a mask is associated with the class name 'person,' the pixel values set to 1 are updated to a higher positive value to increase its significance after consolidation. Conversely, if a mask corresponds to an object, its positive values are updated to large negative values as a penalty before the

TABLE 1. Re-assigning pixel values in segmentation masks.

Previous Values	New Values	Objective
1	-4	This new amplified negative value belongs to a static object or obstacle in the current mask's two-dimensional matrix.
0	0	This neutral pixel does not belong to any classification in the current mask's matrix. It may correspond to free space in the scene, belong to other masks from the batch of prediction results, or represent an object not detectable by the Mask R-CNN model in the input image.
1	4	This new amplified positive value belongs to a person's detection in the current mask's matrix.

final matrix. For all masks, original values set to 0 remain intact.

As a result, every mask associated with an object or obstacle is transformed to exclusively comprise the unique values $\{-4, 0\}$, while any mask corresponding to a person contains only $\{0, 4\}$ as unique values. Incidentally, the length and values in the selected set of integers representing the new pixels may differ in projects with different conditions. However, this set of pixels is suitable for this work, as it characterizes the three derived detection groups of interest for collision-free navigation by a companion robot: Obstacle, Void, and Person.

Following the reassignment of integer pixel values in each instance segmentation mask based on their associated classes, the matrices generated from the prediction results are consolidated into a final two-dimensional structure via matrix addition. This structure preserves the detection-related information, which has been consolidated after the conditional swap of pixel values in the previous step. As the generated masks from the prediction results all have the same shape, the resulting 2D matrix naturally inherits the shape (h, w), where h and w refer to the height and width of the original input frame, respectively. In some cases, generated masks from the prediction results may have overlapping pixels, which can lead to the amplification of numerical values after summing the matrix elements. This can result in values lower than -4 or greater than 4. Although each mask is processed to contain only $\{-4, 0\}$ or $\{0, 4\}$ as a set of unique values, as explained previously, the resulting matrix may contain values in the set $\{-8, -4, 0, 4, 8\}$, owing to overlapping pixels, depending on the content of the original image.

B. DISCOUNTED AVERAGE POOLING

This work proposes an image downsampling algorithm consisting of average pooling steps that are controlled by a recursive function. It calculates the average for every defined patch of the synthesized 2D matrix. The shape of the patches varies at every step of the recursion, depending on the kernels used. These can be viewed as a recommended set of sliding windows holding no weights or kernels having their weights set to 0. In a traditional convolutional neural network, input images are convolved with kernels or filters of shape (F, F) to extract features [29]. The output spatial shape is determined by the padding technique used. Unlike the SAME algorithm, the VALID method uses no padding [30] and typically causes the output size to be lower than the input size, regardless of the stride. As implemented in TensorFlow's module for convolutions [31], the height O_h and width O_w of the output shape can be calculated with the VALID algorithm using standard equations as shown in (1) and (2), where I is the input image, F is the filter, and S is the stride. The subscripts h and w refer to the height and width of the associated elements.

$$O_h = \left\lceil \frac{I_h - F_h + 1}{S_h} \right\rceil \tag{1}$$

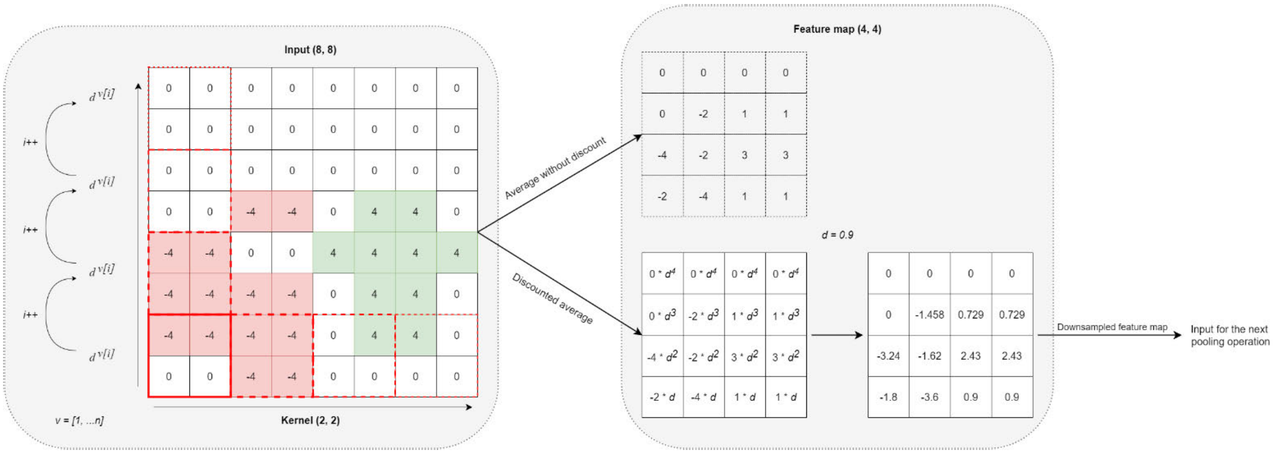
$$O_w = \left\lceil \frac{I_w - F_w + 1}{S_w} \right\rceil \tag{2}$$

The proposed image downsampling algorithm utilizes weightless kernels and follows the VALID convention, thereby avoiding padding. To calculate the average for all non-overlapping patches of shape (K, K) across each input image, the symbol K is used instead of F in subsequent equations and illustrations. The vertical and horizontal strides are also set to K. To ensure that the height and width of the downsampled feature maps in ensuing steps are multiples of K, the maximum number of rows or columns of pixels dropped from the first step of the recursive pooling operation for each image is limited to $K - 1$. This eliminates the need for any further cutbacks. The proposed angular position estimation algorithm, referred to as APEA, comprises three kernels of varied sizes by default: (8, 8), (4, 4), and (2, 2). It must be noted that the weightless kernels are not intended to learn specific features of an image at a low level.

For a companion robot to navigate safely and follow people in unfamiliar environments, it must be highly aware of obstacles that may obstruct its path, particularly those at ground level. Conversely, obstacles that are elevated or farther away pose less of a threat and should carry less weight in the robot's navigation decision-making process. The robot is a wheeled agent that maintains ground contact and cannot fly; hence a discount factor is introduced to gradually decrease the importance of different regions of the feature maps. The impact of the discount value gets boosted linearly at each vertical stride.

At each step of the pooling operation, kernels slide across the input matrices starting from the bottom-left, adapting the algorithm to the problem. Fig. 2 illustrates the algorithm's inner workings, where a kernel of shape (2, 2) slides from the bottom-left towards the bottom-right and then moves upward to calculate the average values from left to right. This process is repeated until the sliding window reaches the input's top-right region, resulting in 16 calculated average values stored correspondingly in a smaller shape matrix (4, 4).

The illustration features the sequence of vertical strides, v, representing the possible upward directions from 1 to n. The discount factor, d, is set to a default value of 0.9 and is raised to the power of $v[i]$ when the kernel makes a stride up. In this example, the discount decreases steadily from 0.9 to 0.6561.



Input (8, 8)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	-4	-4	0	4	4	0
-4	-4	0	0	4	4	4	4
-4	-4	-4	-4	0	4	4	0
-4	-4	-4	-4	0	4	4	0
0	0	-4	-4	0	0	0	0

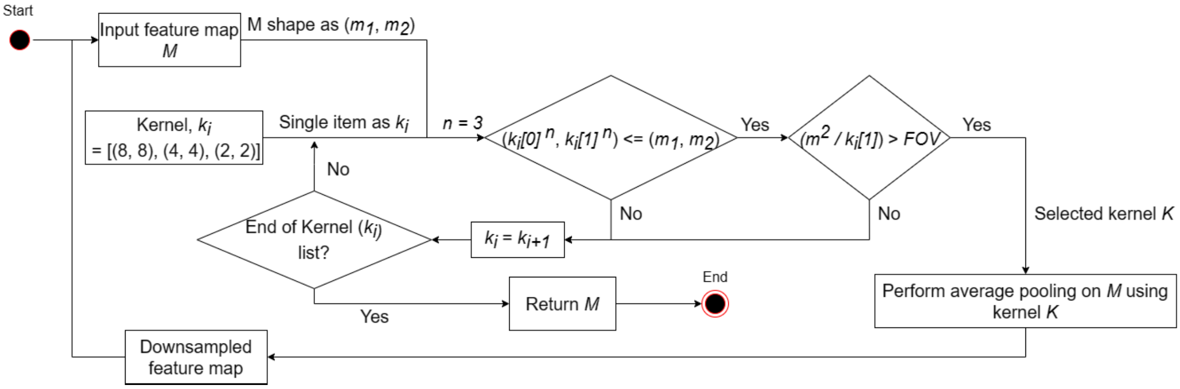
Feature map (4, 4)

0	0	0	0
0	-2	1	1
-4	-2	3	3
-2	-4	1	1

$0 * d^4$	$0 * d^4$	$0 * d^4$	$0 * d^4$
$0 * d^3$	$-2 * d^3$	$1 * d^3$	$1 * d^3$
$-4 * d^2$	$-2 * d^2$	$3 * d^2$	$3 * d^2$
$-2 * d$	$-4 * d$	$1 * d$	$1 * d$

0	0	0	0
0	-1.458	0.729	0.729
-3.24	-1.62	2.43	2.43
-1.8	-3.6	0.9	0.9

FIGURE 2. Average pooling algorithm behavior involving the sliding kernels, the discount factor and feature maps.



Start
Input feature map M
M shape as (m_1, m_2)
Kernel, k_i = [(8, 8), (4, 4), (2, 2)]
Single item as k_i
n = 3
($k_i[0]^n$, $k_i[1]^n$) <= (m_1, m_2)
Yes
(m^2 / $k_i[1]$) > FOV
Yes
No
No
Selected kernel K
No
End of Kernel (k_i) list?
$k_i = k_{i+1}$
Yes
Return M
End
Perform average pooling on M using kernel K
Downsampled feature map

FIGURE 3. Illustration for the kernel selection algorithm, which picks the most suitable kernel to perform the pooling operation recursively.

The matrices shown in (3) and (4) below demonstrate how an input I would be processed by the algorithm represented by the parameterized function $f(I, K, D)$ using a kernel K of shape (2, 2) in conjunction with a default discount factor D of 0.9. It is important to note that input images would typically be much larger, and a dynamic kernel selection process would be required to optimize the downsampling operation, as shown in Fig. 3.

$$I = \begin{bmatrix} 0 & 0 & 0 & 4 \\ 0 & 4 & 4 & 4 \\ 0 & 4 & 4 & 4 \\ 0 & 4 & 4 & 4 \\ -4 & -4 & -4 & -4 \end{bmatrix} \tag{3}$$

$$f(I, K, D) = \begin{bmatrix} 1.62 & 3.24 \\ -0.9 & 0 \end{bmatrix} \tag{4}$$

C. DYNAMIC KERNEL SELECTION

Effectively implementing a downsampling operation requires balancing rapid downsampling and information preservation in image processing. Using the default kernel of shape (8, 8) exclusively may expedite downsampling but could potentially result in significant information loss. Conversely, relying solely on the smallest available kernel may necessitate excessive iterations. The APEA incorporates a kernel selection algorithm employed at each recursive pooling operation step to address this challenge, as depicted in Fig. 3.

A coefficient n is introduced to determine the most appropriate kernel during image processing, and the robot's camera's field of view FOV is utilized. Extensive testing has recommended using a coefficient value of $n = 3$ for optimal performance. The resulting feature map is assessed, and once no more kernels can fit, the column-wise sum is computed. This result determines the specific areas of the scene that the robot must avoid, from left to right. This work maintains the camera's field of view at 58° horizontally during the development and evaluation.

D. 1D MATRIX PROCESSING

To accurately represent the mobile robot's rotation range in degrees based on its field of view for tracking, the column-wise sum operation generates a one-dimensional (1D) matrix from the final feature map. The latter is the final output from the discounted average pooling operation. The central point of view of the robot is indicated by the middle of the resulting range, which is zero degrees, as illustrated in (5). In the last step, a normalized version of the 1D matrix is utilized to obtain a description of the scene, which becomes more comprehensible when plotted on a graph.

As depicted in Fig. 4, high data points on the Y-axis correspond to areas in the input image with the most substantial human presence, while low data points indicate the presence of obstacles to avoid. Data points located at lower positions are more likely to accurately reveal the presence of an object close to the robot or a large obstacle.

In a way, the plotted data represents the robot's perception of its environment, which is then rolled back into a panoramic view from which the angular position of discerned entities can be estimated. Specifically, the plotted data provides an analog representation of the robot's determinable rotation range, which can be used to identify and avoid obstacles while tracking the human subject.

$$r = \left[\frac{-FOV}{2}, \frac{FOV}{2}\right] \tag{5}$$

In Fig. 4, a person is standing on the far left of the camera's view in an office room with poor lighting. The Mask R-CNN framework detects the person and objects, including office chairs, through a camera with a 58° horizontal field of view. The discounted average pooling algorithm processes the output, resulting in a normalized one-dimensional matrix containing high data points for the target person and dips for the recognized obstacles.

This data is mapped to the robot's camera field of view, where the middle coordinate on the X-axis represents the 0-degree angle from the robot's perspective. The 0 coordinate represents the far edge of the robot's perception on its left through the camera, without rotating, and the same is true for the maximum value on the plot's X-axis and the right edge of the captured image frames. The range of coordinates on the X-axis correlates with the range of degrees r in (5), making this association practical.

Fig. 4 demonstrates the mapping process in which the APEA produces a negative rotation angle of −22.97° by matching the length of the one-dimensional matrix, consisting of 77 data points, with the horizontal field of view of an Asus Xtion Live Pro camera, set to 58° [27].

The precise rotation angle for the robot to focus on the target person can be calculated in degrees and radians using the equations (6) and (7), respectively, where x denotes the index of the highest value (or data point) in the 1D matrix, and r is the range of the matrix, i.e., the number of data points depicting the scene at any moment. Additionally, the FOV variable indicates the camera's horizontal field of view.

$$deg = \left(x \cdot \frac{FOV \cdot 0.5}{r \cdot 0.5}\right) - FOV \cdot 0.5 \tag{6}$$

$$\begin{aligned} \text{rad} = &\left(\frac{\text{x} \cdot \text{FOV} \cdot 0.5}{\text{r} \cdot 0.5}\right) \cdot \left(\frac{\pi}{180}\right) \\ &- (\text{FOV} \cdot 0.5) \cdot \left(\frac{\pi}{180}\right) \end{aligned} \tag{7}$$

TABLE 2. Simulation experiment results.

Testing component	Evaluation results
1D matrix length (Total data points)	80
1D matrix mean value	0.53339
1D matrix median value (Search threshold)	0.50941
Angle A	7.98°
Angle B	82.02°
Angle C	90°
Congestion ratio – left (c_1)	0.413
Congestion ratio – right (c_2)	0.605
Final feature map shape	(60, 80)
Image input shape	(480, 640)
Index of highest data point	53
Mapping quotient	8.0
Recommended rotation angle r	9.425°
Recommended side for obstacle circumvention	Right
Region of interest – Index range per exploration algorithm	[51, 79]
Region of interest – Mapped index range (to original size)	[408, 632]
Region of interest – Sampled pixels	10752
Region of interest – Sampling rate (pixels)	10%
Region of interest – Extract shape	(480, 224)
Total new feature maps	3

V. PATH PLANNING WITH DYNAMIC CONTROL POINTS

The target person may become partially occluded during the tracking and following activity, leading to mixed signals for the robot's immediate actions based on the plottable APEA output. An exploration algorithm is introduced to search for the boundaries of the target person in the normalized 1D matrix to mitigate collision risks. Additionally, the algorithm explores the edges of objects recognized in the scene and recommends a path to take, either from the left or right, to circumvent obstacles as necessary. By selecting the maximum between the median and mean of the 1D matrix's values as a numerical threshold, the algorithm distinguishes between areas from the scene with strong human presence and objects, inferring partial occlusion. This approach enables the detection of boundaries for the target person and the selection of data points in the 1D matrix that denote regions of interest in the input image from the robot's perspective. In addition to facilitating the comparison of the consolidated mask matrix with the pixel-distance matrix, the plottable 1D matrix can be mapped to the original input's shape. This mapping requires obtaining the quotient indicating how much the final feature map of the discounted average pooling algorithm was reduced in size compared to the original input. The extracted fragment of pixel values from the mapped matrix is designated as the region of interest's mapped index range, presented in Table 2, Table 3, and Table 4 in $[y, x]$ format. Partial occlusion can be detected if samples from a region of interest containing the target person also have pixels corresponding to objects. The number of pixels in the mapped range is limited to a 10% sampling size to facilitate continuous processing at several frames per second and optimize path planning. This pixel selection involves randomly selecting N rows and columns

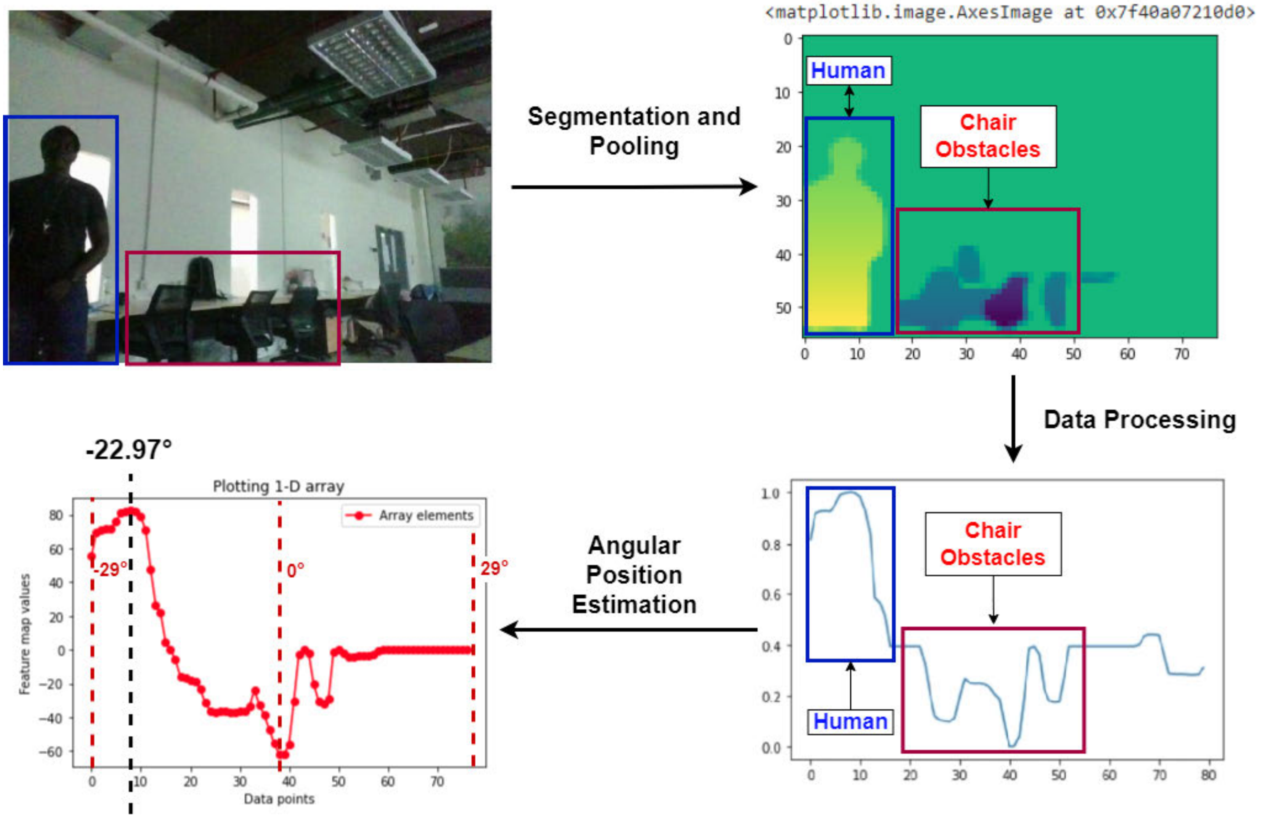


FIGURE 4. Demonstration of the APEA process using images and graphs to represent the sequential steps in a clockwise direction.

from the slice, where N is a tenth of the product of the slice's width and height. The quotient Q, slice's shape S, and sample size N are calculated using the consolidated mask's width (M_w), the length L of the plottable 1D normalized array, the left and right boundaries (B_l and B_r) returned by the exploration algorithm, and the slice's height and width (S_h and S_w). The obtained values, as demonstrated via equations (8), (9) and (10), allow for the extraction of the region of interest after running the exploration algorithm for depth data fusion.

$$Q = \frac{M_w}{L} \tag{8}$$

$$\mathrm{S} = (\lfloor B_l \cdot Q \rfloor , \lfloor B_r \cdot Q \rfloor) \tag{9}$$

$$N = \left\lfloor \frac{(S_h \cdot S_w)}{10} \right\rfloor \tag{10}$$

The proposed Dynamic Path Planning (DPP) technique is a collision prevention approach that enables a robot to navigate a scene while avoiding obstacles and humans. DPP considers the overall congestion of entities in the scene from the robot's perspective. It then assigns a decimal value between 0 and 1 as a ratio to measure the collision risk on either side of the robot. Specifically, the congestion perceived by the robot on its left and right sides is denoted as c_1 and c_2, respectively. DPP calculates mean values from the normalized 1D array for the robot's two sides to determine the congestion values. It computes the mean between the array's zero index and middle index and the mean from the middle index to the last. A low value indicates the potential presence of obstacles to avoid on the corresponding side, while a high value suggests either free space or strong human presence.

Despite the output from previous stages of the technique, there is no guarantee that the target person's position in the scene is directly reachable by the robot. For example, the robot's right side may contain obstacles, even if the human presence is significant. In such cases, it may be more favorable to consider forming a path slanted towards the left side first for circumvention purposes before joining the target on the right side.

This precautionary collision prevention technique warrants new research work; nonetheless, DPP introduces what could be part of the underlying computations for such an approach.

A. REGION OF INTEREST EXTRACTION

The process of ROI extraction has been previously discussed, as it plays a critical role in controlling the movement of the mobile robot. This paper's subsection presents two photographs. One captures an indoor scene of a child playing; the other shows an outdoor scene of a man standing near identifiable objects. These images demonstrate the functionality of the ROI extraction technique. This technique's reliability is expected to remain constant regardless of the height at which a scene is captured, whether from ground level or human eye level. This convenience is due to its dependence on the stable

TABLE 3. Simulation experiment 2 results.

Testing component	Evaluation results
1D matrix length (Total data points)	80
1D matrix mean value	0.29450
1D matrix median value (Search threshold)	0.16627
Angle A	16.67°
Angle B	73.33°
Angle C	90°
Congestion ratio – left (c_1)	0.262
Congestion ratio – right (c_2)	0.326
Final feature map shape	(60, 80)
Image input shape	(480, 640)
Index of highest data point	71
Mapping quotient	8.0
Recommended rotation angle r	22.475°
Recommended side for obstacle circumvention	Right
Region of interest – Index range per exploration algorithm	[65, 79]
Region of interest – Mapped index range (to original size)	[520, 632]
Region of interest – Sampled pixels	5376
Region of interest – Sampling rate (pixels)	10%
Region of interest – Extract shape	(480, 112)
Total new feature maps	3

TABLE 4. Simulation experiment 3 results.

Testing component	Evaluation results
1D matrix length (Total data points)	80
1D matrix mean value	0.33341
1D matrix median value (Search threshold)	0.33017
Angle A	5.08°
Angle B	84.92°
Angle C	90°
Congestion ratio – left (c_1)	0.23783
Congestion ratio – right (c_2)	0.42900
Final feature map shape	(60, 80)
Image input shape	(480, 640)
Index of highest data point	63
Mapping quotient	8.0
Recommended rotation angle r	16.67°
Recommended side for obstacle circumvention	Right
Region of interest – Index range per exploration algorithm	[59, 66]
Region of interest – Mapped index range (to original size)	[472, 528]
Region of interest – Sampled pixels	2688
Region of interest – Sampling rate (pixels)	10%
Region of interest – Extract shape	(480, 56)
Total new feature maps	3

APEA output. As demonstrated in Fig. 5, the exploration algorithm leverages a threshold on the APEA output to extract ROIs (Regions of Interest) that contain the target person.

Furthermore, the ROI extraction technique is designed to avoid space that contains non-human elements, except for free space around the target person, as depicted in both Fig. 5 and Fig. 6. Here, the free space marks the area within the robot's perceived environment that can be considered during navigation to follow the human target, in case of partial occlusion or blocked path. These examples prove that the proposed ROI extraction technique is effective for indoor and outdoor environments, regardless of the scene's congestion.

B. TRIANGULATION ALGORITHM

The law of sines [32] calculates the angles necessary for safe navigation toward the target person using the congestion ratios c_1 and c_2. This principle is shown in (11) and (12). The exploration algorithm identifies the favorable area to consider and estimates the angles of obstacles blocking the robot's path to the target person. The algorithm also identifies regions of interest containing the strongest human presence for further processing with depth information. DPP uses fundamental trigonometry formulas to estimate safe paths for robot navigation. As shown in Fig. 7, a clear straight path from the robot to the target person is initially assumed; however, an obstacle prompts the creation of a new tilted path forming the hypotenuse of a virtual triangle. This new trajectory generates the three angles required to solve the triangle with the sine rule. The calculation of angle A in (13) is similar to the degree value computation by the APEA in (6). However, (13) uses the angle of the obstacle's edge, represented by theta (θ), instead of the camera's field of view FOV. Angle B is the difference between 180° and the sum of angles A and C, where angle C is pre-set to 90°.

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} \tag{11}$$

$$\frac{\sin A}{a} = \frac{\sin B}{b} = \frac{\sin C}{c} \tag{12}$$

$$\begin{aligned} A &= \left(x \cdot \frac{\theta \cdot 0.5}{r \cdot 0.5}\right) - \theta \cdot 0 \\ B &= 180 - A - C \\ C &= 90 \end{aligned} \tag{13}$$

Side b of the virtual triangle is assigned the average distance of the obstacle causing partial occlusion, obtained through pixel and depth data fusion. The value of side b is computed by dividing the sum of obstacle-related pixel values x_i from the depth camera topic in ROS by the total amount of pixels n, as shown in (14). Side a is calculated using angle A in radians and the sine rule, while the value of side c is obtained using the Pythagorean theorem [33].

$$\begin{aligned} b &= \frac{1}{n} \sum\nolimits_{i=1}^{n} x_i \\ a &= b \cdot \frac{\sin A}{\sin B} \\ c &= \sqrt{\alpha^2 + b^2} \end{aligned} \tag{14}$$

The DPP method prioritizes angles over distance values for path planning, allowing the robot to prioritize areas based on visual input when navigating toward the human target in the environment. This iterative navigation process results in high accuracy in a simulation under various scenarios. The robot continuously collects fresh image input to re-analyze the scene and perform path re-estimation. Navigation activity

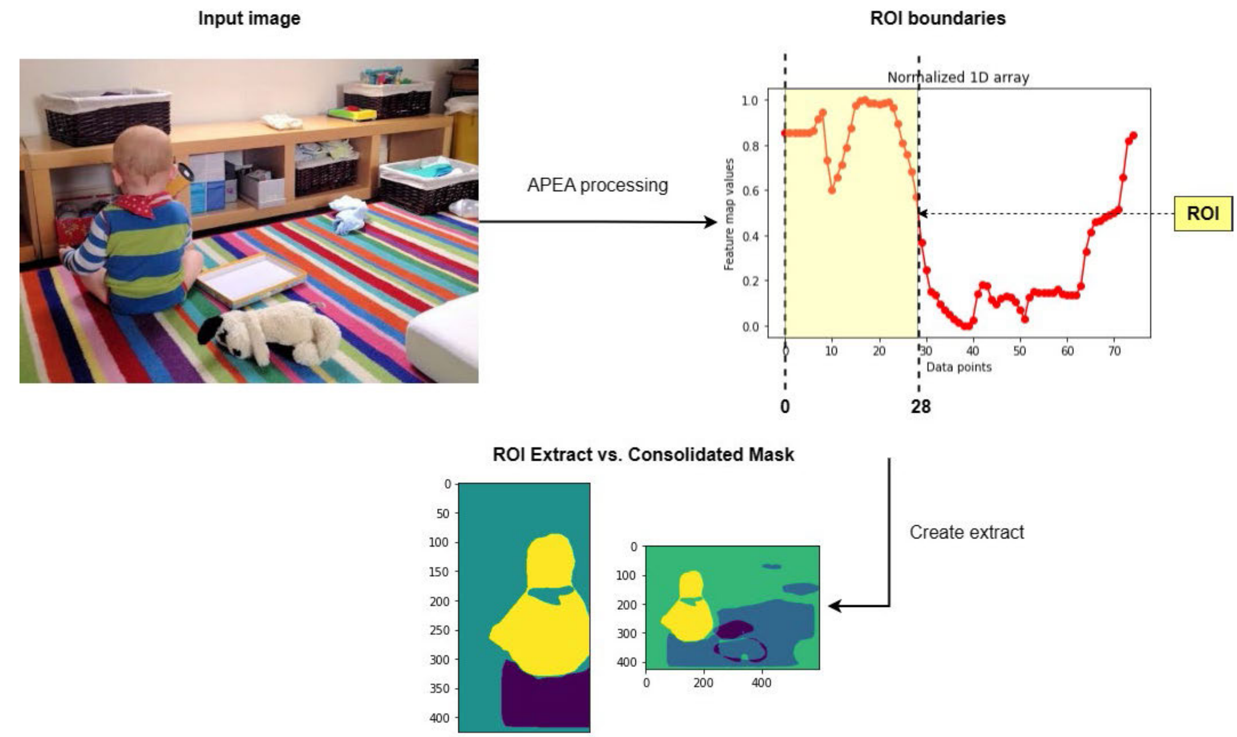


FIGURE 5. Successful ROI extraction (first example). This illustrates areas of the perceived scene that interest the robot for safe navigation.

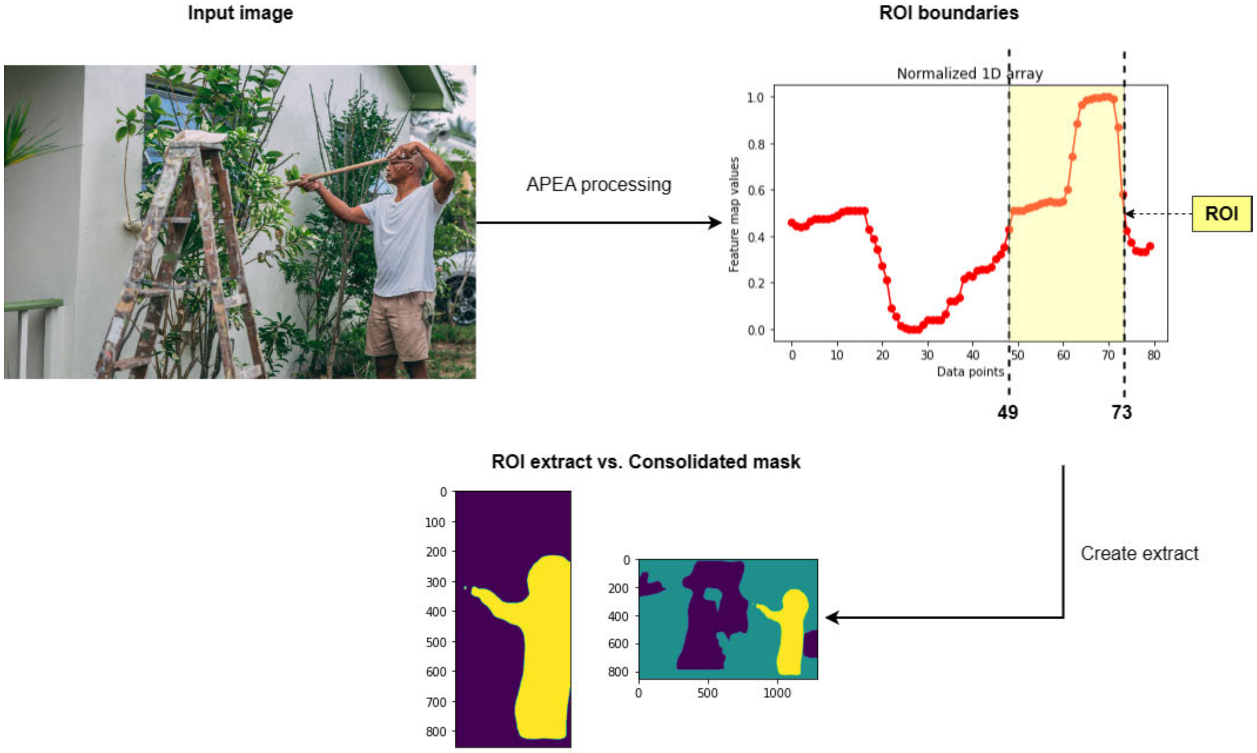


FIGURE 6. Successful ROI extraction (second example). Again, the area of the scene featuring the target person and free space gets selected.

can be demarcated by frames per second or time steps using a new parameter t, representing a coordinate in the environment at a given time along the path P influenced by control points.

This robust obstacle circumvention logic uses the path tracing equation for a standard quadratic Bézier curve, as shown in (15), with R representing the robot's position, C a selected

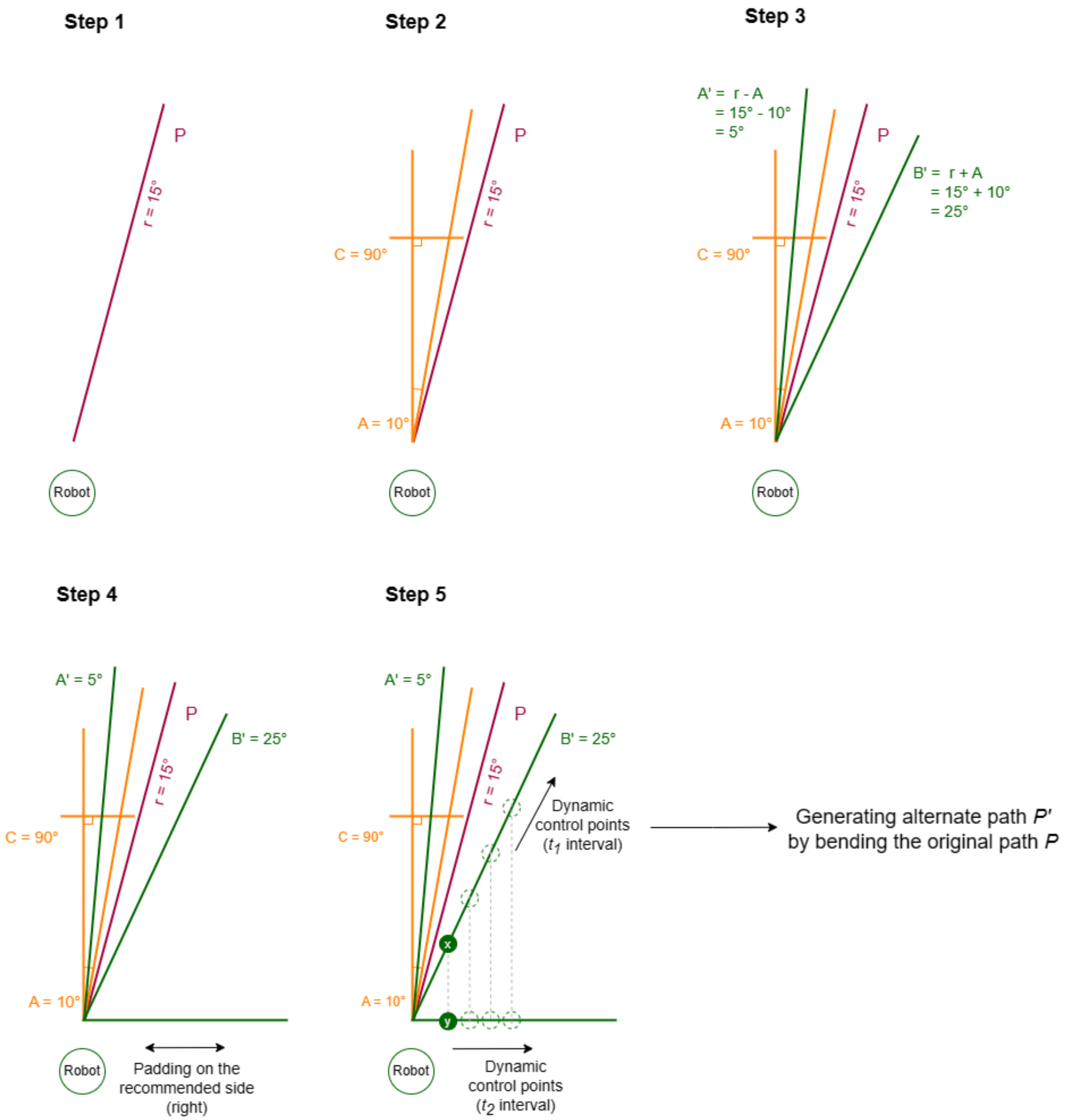


FIGURE 7. Breakdown of the steps to generate dynamic control points for path planning, assuming the target person is first detected on the robot's right side at a 15° angle. Hypothetically, there is an object in front likely to cause a collision if the robot naively follows the straight path P.

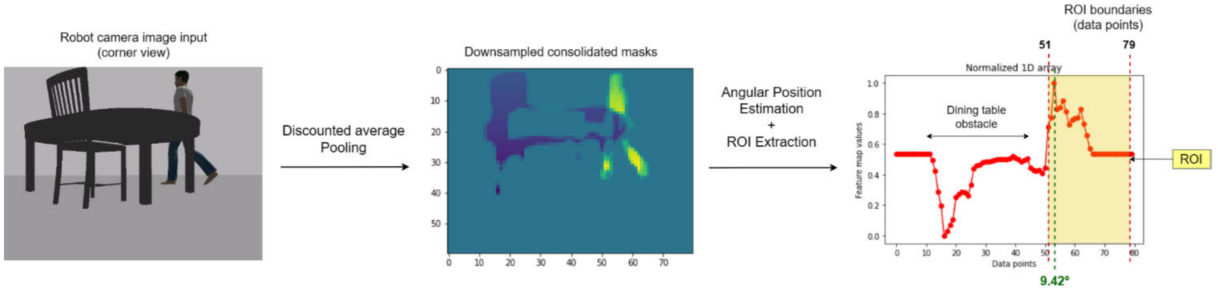


FIGURE 8. Breakdown of the image processing, from the input image to the APEA output and ROI extraction for Experiment 1.

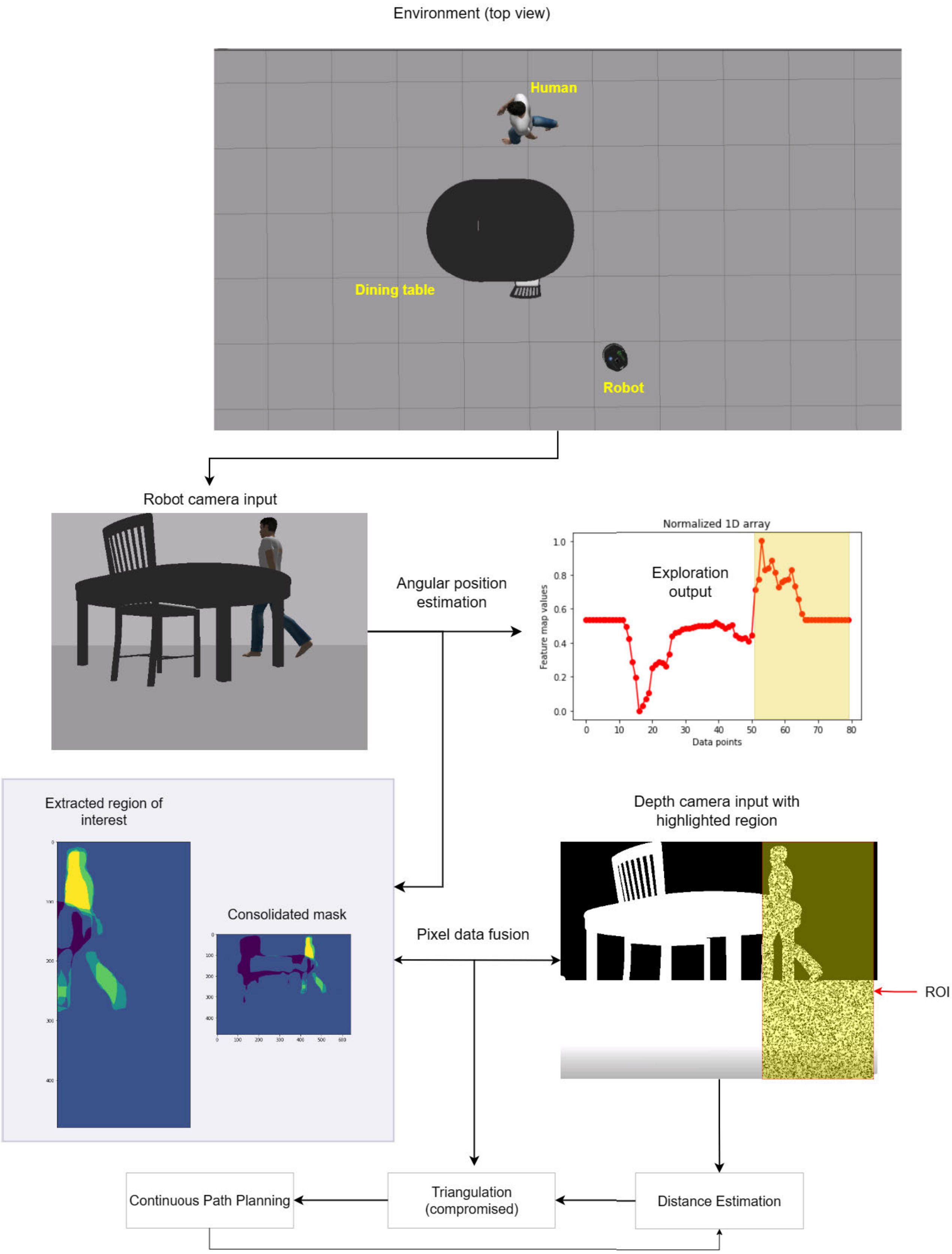


FIGURE 9. Demonstrating how the transformations applied to the image input aids the mobile robot in scene understanding and path planning.

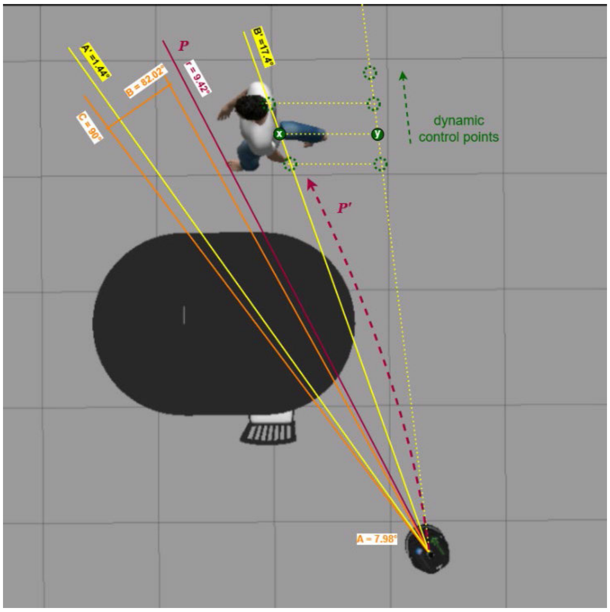


FIGURE 10. Demonstrating how the robot estimates the alternate path P′ for safe navigation and person-following tasks.

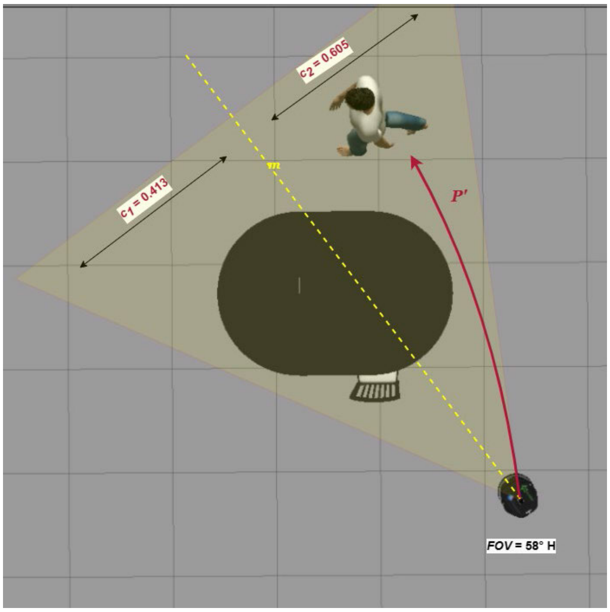


FIGURE 11. Demonstrating how path P′ aids the robot in avoiding obstacles (table and chair) in the same environment as Fig. 10 (image cropped differently).

control point, and H the human target's position in the environment.

$$B(t) = (1-t)^2 \cdot R + 2t \cdot (1-t) \cdot C + t^2 \cdot H, 0 \leq t \leq 1 \tag{15}$$

Dynamic control points are generated in the prototype using the outputs of preceding algorithms in the DPP process flow. Fig. 7 illustrates the addition of angle A's value to both sides of the original path P, based on the rotation angle r from the APEA output, resulting in $A' = r$ - A and $B' = r + A$. Padding is suggested to avoid obstacles in the appropriate direction. The previously indicated parameter t is divided into t_1 and t_2, representing intervals on their respective trajectories. These independent variables may have different values, as illustrated in Fig. 7.

VI. EVALUATIONS

The DPP technique's effectiveness is demonstrated through several scenarios in a simulated environment using a Turtlebot model with a Kobuki base [34] and an Asus Xtion Pro Live 3D sensor with a 58° H, 45° V, and 70° D field of view. The sensor's image resolution is set to 640 × 480, and its distance range extends up to 3.5 meters. However, object choices for testing were restricted to those in the model's training dataset due to the limited performance of the instance segmentation model [27]. The simulation tests include a bed, chair, sofa, and dining table. To aid with performance, the experiments were distributed across multiple platforms. Some were conducted on Google Colab, which offers a free NVIDIA®T4 GPU.

This section presents the tabular evaluation results and relevant visual representations of the outcomes. The simulation scenarios were inspired by a 2018 study [2] and were designed to test the navigation prototype using the DPP modules discussed in this paper. The objective is to demonstrate how dynamic control points can be used for collision avoidance and to adjust the mobile robot's path to reach the target person.

A. EXPERIMENT 1: INFERRING SAFE NAVIGATION PATHS

The first scenario depicted in Fig. 8 tests the robot's ability to navigate safely in the presence of a dining table and a chair obstructing its path, with the person model moving away from the robot. The challenge is to avoid misinterpreting table legs or chair legs as the target person's legs. As shown in Fig. 9, the proposed DPP technique leverages Mask R-CNN for detection and classification while extracting meaning from scenes. Although instance segmentation masks may be matched with incorrect labels and object shapes and dimensions may be distorted, the robot can utilize depth camera data to approximate essential obstacle avoidance values by discerning between human and non-human elements without needing specific labeling.

The robot detects the angular position and dimensions of the table and chair to build a suitable path P' for person-following (see Fig. 11), even though the initially-estimated path P (Fig. 10) based on the person's angular position is prone to collision. The evaluation results in Fig. 10 do not include depth values in meters as the DPP technique does not heavily rely on fixed distance values for robot locomotion. Although the virtual triangles' sides are assigned distance values during the path-planning process, they may lack precision or accuracy. Nonetheless, the exploration algorithm provides sufficient information to optimize the triangulation algorithm's values, as reflected in Fig. 14. This is later reiterated in another experiment in Fig. 18.

Fig. 10 depicts the successful application of the proposed DPP modules, which integrate depth data, for the robot's

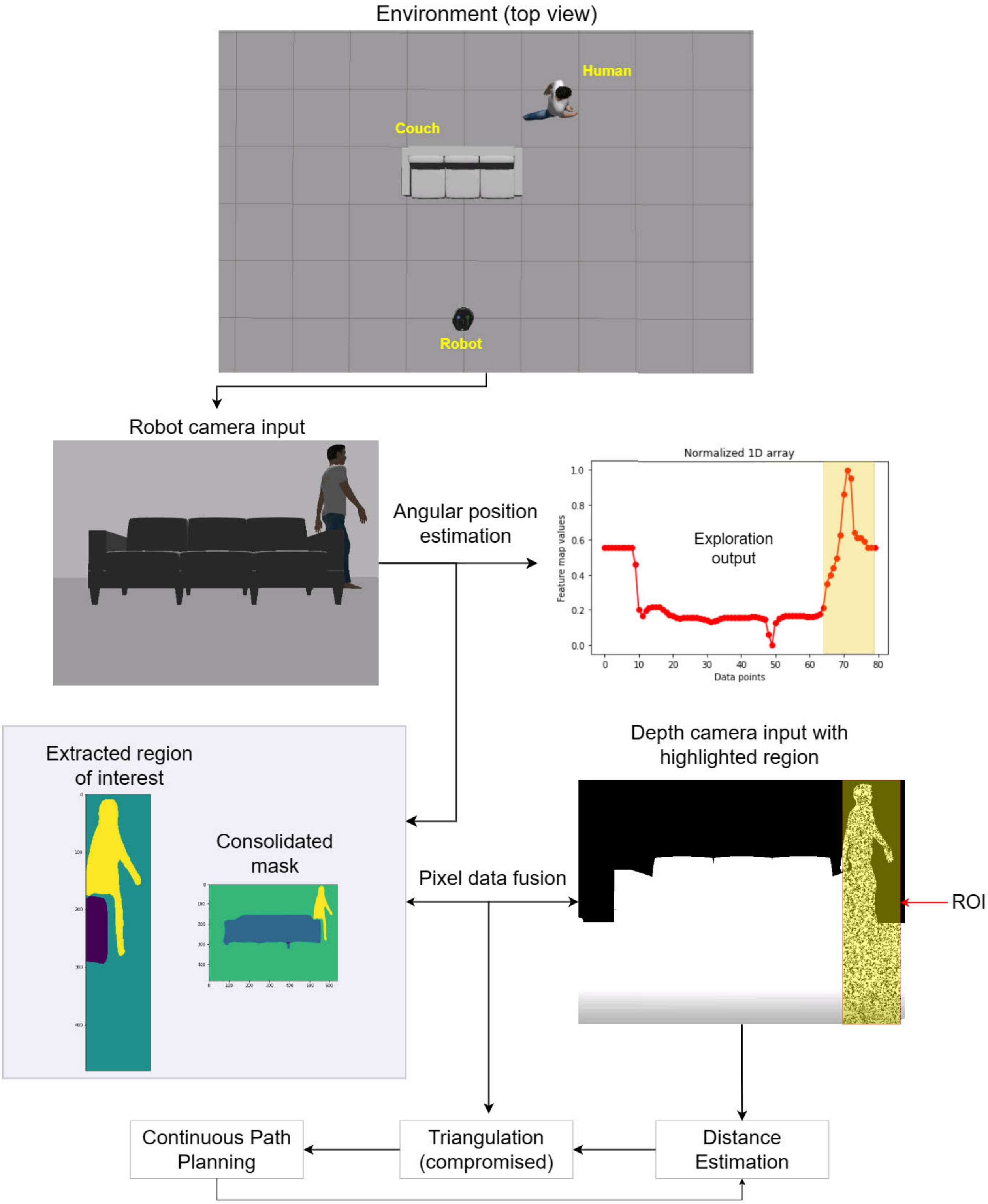


FIGURE 12. Breakdown of the processing for Experiment 2, from the camera input to the path planning computation.

environment analysis and trajectory prediction based on RGB image frames. The results demonstrate the high accuracy and efficiency of the DPP technique in ensuring safe path planning.

B. EXPERIMENT 2: DETECTING OBJECTS AND DIMENSIONS

The second experiment, presented in this paper, features a human model on the right side of the input image, standing

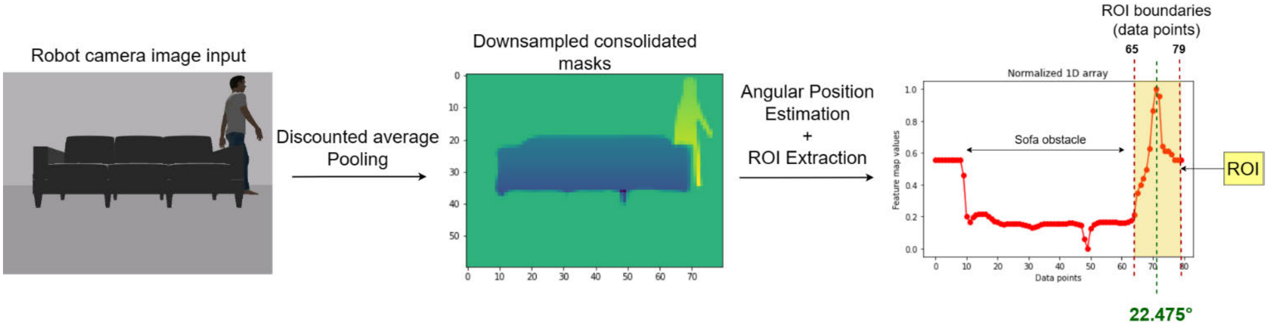


FIGURE 13. Precise segmentation and ROI extraction let the robot understand the scene and make accurate navigation decisions.

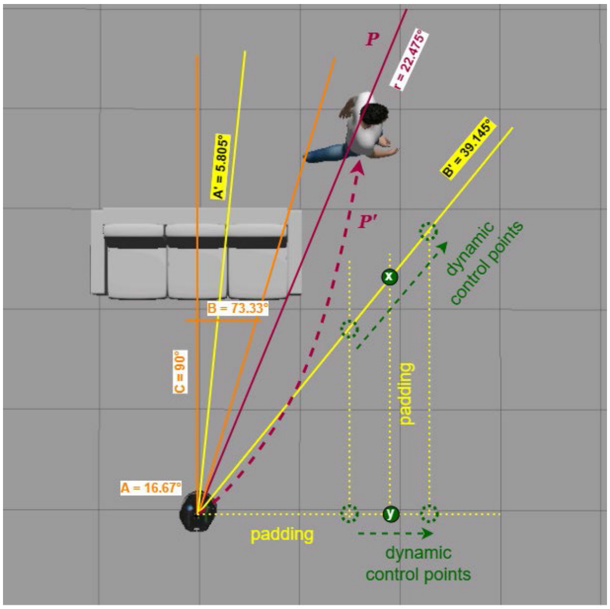


FIGURE 14. Visualization for the robot's understanding of the scene and path planning estimations. This figure is a cropped version of the environment from Fig. 15.

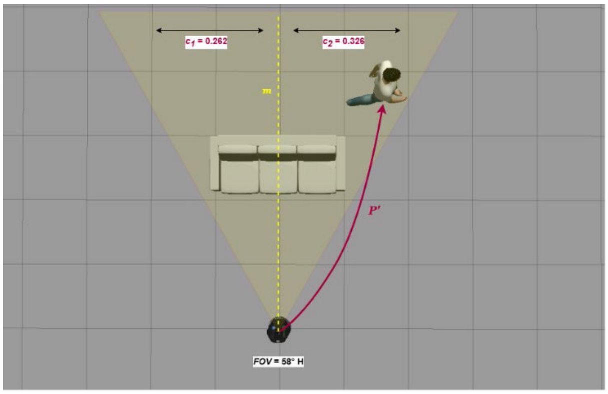


FIGURE 15. Congestion ratios and Path P' for Experiment 2, where c_1 is 0.262, and c_2 is 0.326. This figure is the same environment as Fig. 14.

behind a large couch—an obstacle to the robot's movement. The sofa is approximately two meters away from the robot and dominates its field of vision. Furthermore, there may be confusion about which legs belong to the person and which belong to the couch, particularly on the right side. However, despite these challenges, Mask R-CNN accurately deconstructs the input image by providing the appropriate set of masks, which are then fed into the APEA process for ROI extraction. The resulting ROI (depicted in Fig. 12) contains object pixels, and the depth camera validates the presence of an obstacle in the extracted area, signaling the robot to navigate around the couch by planning a new path based on the values generated from the scene (see Fig. 12, Fig. 13, and Table 3).

Fig. 14 illustrates how the dynamic control points are continually updated during the robot's navigation to avoid obstacles. The dashed path P' shown in the figure depicts an alternative version of the original unsafe path P, which was initially estimated through the APEA output. By updating the control points dynamically, the robot can generate a new safe route that circumvents any obstacles in its path. This capability allows for more efficient and safe navigation in complex environments.

C. EXPERIMENT 3: NAVIGATING WITH PARTIAL OCCLUSION

Fig. 16 and Fig. 17 present the third simulation experiment, consisting of a tilting bed and a standing human model positioned behind it, both of which are observable to the mobile robot. This setup represents a case of partial occlusion, and the aim is for the robot to navigate toward the target person by utilizing the depth camera input and APEA output. The simulated environment reveals that the robot necessitates further processing, as blindly pursuing the straight path P, depicted in Fig. 18—based on the plottable 1D normalized array—would result in a collision with the bed. Therefore, the robot must determine a secure path P' by considering the outcomes of the exploration and triangulation algorithms and the calculated dynamic control points.

The APEA's downsampled feature map exposes Mask R-CNN's inaccurate bed set size estimates. However, the region of interest extraction and depth camera data suggest the presence of obstacles. The robot responds by generating control points along multiple trajectories to alter its path from P to P', avoiding collisions. Table 4 presents the recursive pooling algorithm's output, creating a feature map of 80 data points, denoting the angular position and object presence in

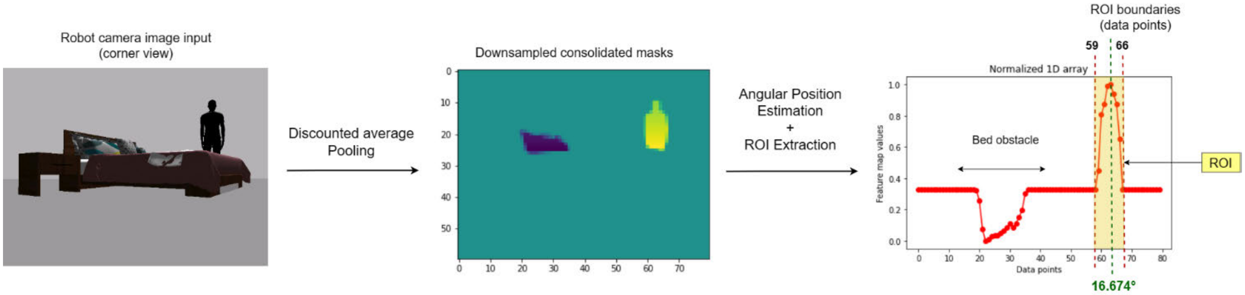


FIGURE 16. APEA output for Simulation Experiment 3, which exposes Mask R-CNN's inaccurate bed set size estimates.

the input image. The robot is initially recommended to rotate clockwise by 16.67° to optimize obstacle avoidance. This approach offers a promising solution for navigating mobile robots in complex, cluttered environments, integrating depth camera data, region of interest extraction, and recursive pooling algorithms.

Fig. 16 depicts the estimation of the angular position of the target person after the robot's camera captures the scene. Identifying partial occlusion triggers the extraction of the ROI, which prioritizes the perceptible area for safe obstacle avoidance. The exploration and triangulation algorithms utilize the consolidated mask and depth camera data to generate accurate angles for collision-free navigation. Specifically, Fig. 17-19 demonstrate the successful outcome of this process.

The values presented in Fig. 18 result from the DPP modules responsible for determining angular positions, ROI extraction, and triangulation. All lines in the diagrams presented in the Evaluations section are drawn using precise angles for clarity. Fig. 19 assesses the congestion ratios (c_1 and c_2) on both sides of the robot's vision field (divided by the dashed line m), which guides the robot's decision to prioritize its right side when circumventing the bed set ($0.429 > 0.237$).

D. EXPERIMENT 4: YOLOv8 AS THE SEGMENTATION MODEL

The system's modularity ensures easy maintenance and upgrades for enhanced accuracy and performance. The current prototype employs Mask R-CNN as the instance segmentation model, as explained in this work's approach overview and prototype description. However, recent state-of-the-art detection models merit consideration, given the significant segmentation inaccuracies observed in Experiment 3. While depth data integration allows for the avoidance of incorrectly measured obstacles in the robot's path, a better-trained segmentation model must be applied to the input camera frames to increase efficacy throughout the iterative DPP process.

In this experiment, Ultralytics' YOLOv8 is used instead of Mask R-CNN as the model generating the masks for the APEA and ROI extraction. Built upon previous YOLO versions, YOLOv8 is a faster, more capable candidate than its predecessors, which showed superiority to Mask R-CNN [35]. Table 5 offers additional information on the specific segmentation model employed for this next experiment.

TABLE 5. YOLOv8 segmentation model specs.

Specs	Values
Size (pixels)	640
mAP^{box} 50-95	44.6
mAP^{mask} 50-95	36.8
Speed – CPU (Central Processing Unit) ONNX (ms)	155.7
Speed – A100 Tensor RT (ms)	1.47
Parameters (M)	11.8
FLOPs (B)	42.6

The Pedestrian Direction Recognition dataset from the University of Alicante [37] was used to evaluate the accuracy and performance of the APEA and ROI extraction algorithms, with YOLOv8 as the segmentation framework. The extracted dataset contains 54,731 image frames depicting pedestrians walking alone or in groups in various directions. From this total, 1,400 images containing single pedestrians from different scenes were selected and arranged sequentially. The evaluation utilized a modified version of the APEA algorithm powered by YOLOv8 and the same GPU and CPU used in the previous experiments. The obtained results shed light on the effectiveness of the proposed technique for person detection and tracking.

The YOLOv8 algorithm demonstrated high person detection performance, achieving 100% accuracy on the raw pedestrians' dataset and accurately identifying objects reflected in the COCO dataset on which it was trained. The algorithm's output was then utilized by the APEA to compute the target's angular position and to perform ROI extraction, successfully defining areas with minimal obstacles or maximum free space that featured the target person. The average inference time for YOLOv8 was recorded at 11.3ms, while the image processing algorithms from the DPP prototype exhibited an average time of 15.6ms. This demonstrates that the image processing component of the DPP prototype is highly adaptable and possesses significant potential for future optimization.

Such capabilities are highly beneficial in real-life applications, allowing robots to comprehend areas within captured scenes and enabling advanced path planning, even in cases of partial occlusion. Unlike naïve bounding boxes from popular object detection models, the ROI extractions present a

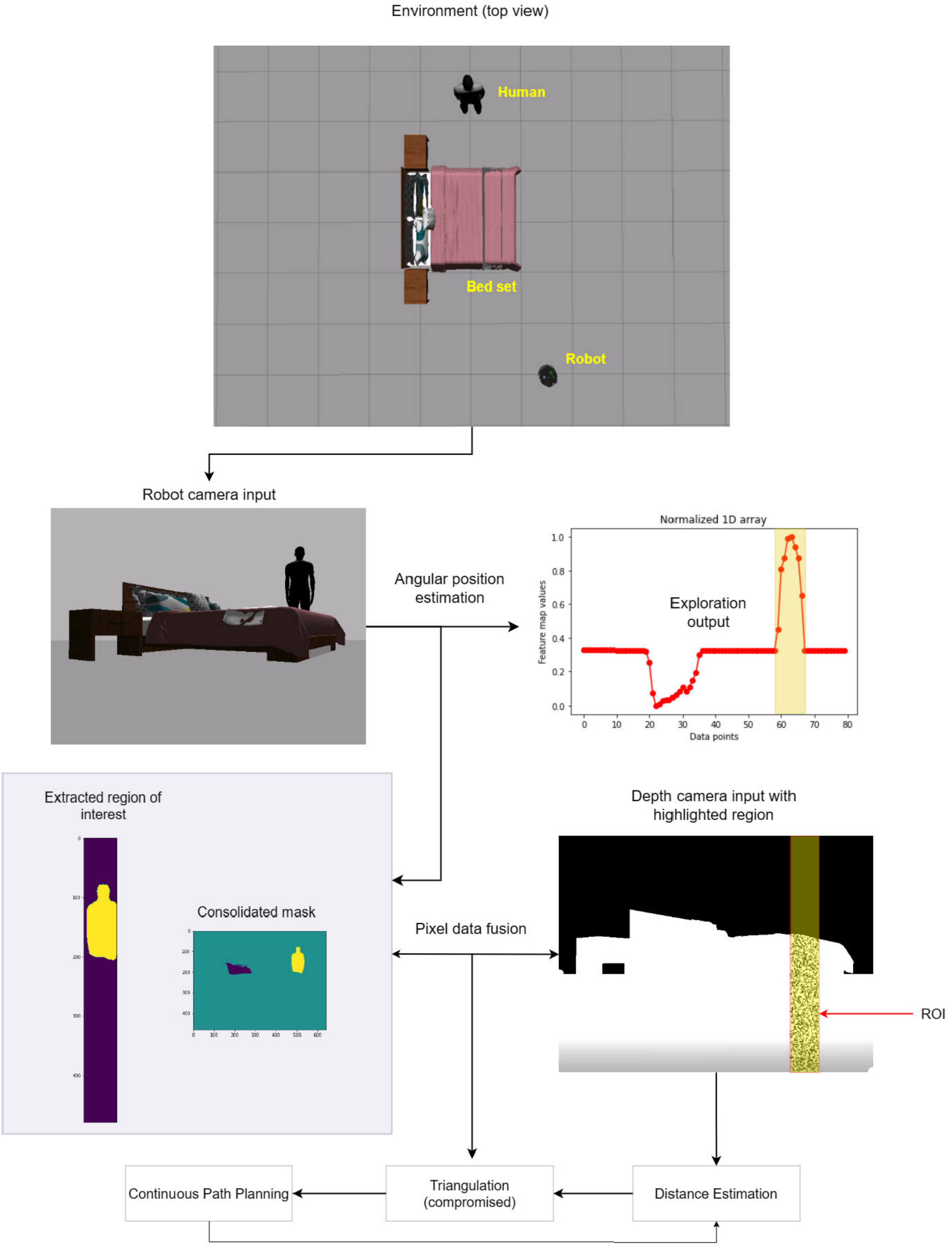


FIGURE 17. A look at the DPP modules in action for Experiment 3, which shows the imperfect segmentation masks and data fusion.

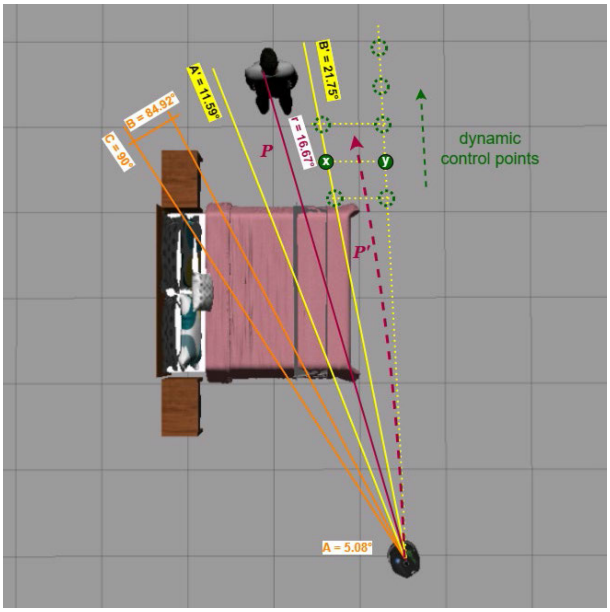


FIGURE 18. Estimating the alternate path P′ for Experiment 3.

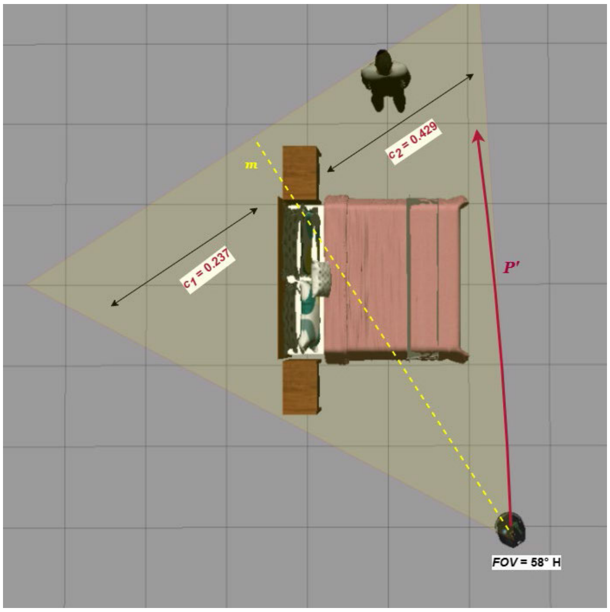


FIGURE 19. Visualizing the congestion ratios c_1 (0.237) and c_2 (0.429), hinting at the safety levels for navigation on the robot's left and right. This is the same environment as Fig. 18, but cropped differently.

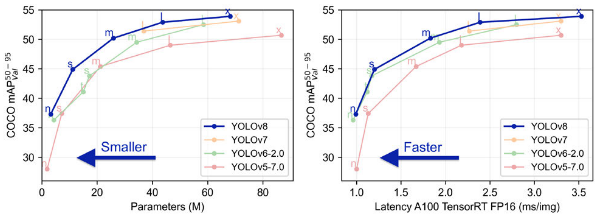


FIGURE 20. Architecture and performance comparison between YOLOv8 and previous YOLO models [36].

cleverer approach, highlighting areas in the scenes where the target is present and where the robot has minimal collision risks.

One notable advantage of this technique is its potential to resolve the challenging issue of the search space in Centroid Tracking, which depends on the Euclidean distance between the centroid of a detected entity across consecutive frames in a video. By utilizing ROI slices that incorporate spatial data about a target person's angular position and safe surroundings, this information can be integrated into the Centroid Tracking algorithm to facilitate straightforward camera-based motion prediction and target recognition. Nonetheless, further exploration and testing of this approach may be necessary, potentially warranting additional research.

VII. CONCLUSION AND FUTURE WORKS

This work designed, developed, and evaluated a deep learning-based companion robot prototype for safe navigation and obstacle avoidance indoors, using only an RGB-D camera as the input sensor. To achieve this, a simulated environment was used to examine a companion robot's path-planning ability using visual data. The experiments involved detecting objects, estimating dimensions, inferring safe paths, and circumventing obstacles. The results presented in Section VI (Evaluations) indicate that the developed companion robot prototype achieved the research objectives. The DPP method presented in this work successfully enabled the person-following robot to perform intelligent curve manipulation for safe path planning to avoid objects in the initial trajectory estimated by the APEA process. Its efficacy is intricately linked to the precision of its underlying segmentation model. As long as the segmentation masks accurately depict the environment, the approach can achieve a 100% understanding of the scenes captured by the robot's camera across frames. This claim is supported by the results of Experiment 4, which utilized a distinct prediction model from the preceding experiments described in Section VI. Overall, the study successfully designed and developed the intended companion robot prototype using input solely from a visual sensor, without having recourse to other modern sensor technologies.

The research demonstrates the potential for future modifications, improvements, and extensions of the proposed segmentation-based dynamic path planning approach. It offers a unique yet straightforward technique for scene understanding using pixel data without the burden of extensive neural network configuration. The suggested method uses instance image segmentation and elementary matrix calculations to aid a robot in identifying the angular position of elements in its surroundings, with resilience to various environmental conditions. Additionally, it combines visual and depth information for scene understanding and path prediction with little computing cost, thanks to a series of novel image processing and data fusion techniques.

The general nature of the solution, with its modular architecture and flexibility, makes it a promising candidate for further development and refinement, potentially contributing to the advancement of various areas, such as self-driving automobiles and assistive technologies for the visually impaired.

DPP also has potential implications for various applications demanding low-cost solutions, including warehouse automation, medical robotics, and search and rescue missions. Nonetheless, further research is necessary to explore the potential of this technique in more dynamic and unpredictable environments.

The next step towards improving robot navigation and continuous person tracking is designing and developing a novel model for multi-sensor data fusion to address the existing limitations in the robot-person following domain, particularly in dynamic indoor environments without predefined maps. This follow-up approach will involve a depth camera to obtain 3D information about the robot's surroundings. It will also include precisely positioned sensors, preferably ultrasonic sensors, to detect static and moving obstacles at close range and a motion-tracking component to predict the target person's movements while comprehending the trajectory of moving entities in a scene. A memory layer holding the fused data could help plan the robot's path from its recent course and maintain a visual representation of the environment. This layer will assist in reconstructing the route the target person takes in case of severe occlusion.

Moreover, indoor ceiling lights and other regular elements are natural landmarks for self-localization. Thus, the companion robot could rely on the map it has built throughout its navigation and person-following task in case of low lighting or the absence of artificial illumination from ceiling lights. For intelligent path planning, the idea is to blend that information with recent observations of the target person's pose to handle occlusions while tracking them and navigating autonomously.

REFERENCES

[1] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020, doi: 10.1109/ACCESS.2020.2975643.

[2] M. T. K. Tsun, B. T. Lau, and H. S. Jo, "An improved indoor robot human-following navigation model using depth camera, active IR marker and proximity sensors fusion," *Robotics*, vol. 7, no. 1, p. 4, Jan. 2018, doi: 10.3390/robotics7010004.

[3] R. Barber, J. Crespo, C. Gómez, A. C. Hernámdez, and M. Galli, "Mobile robot navigation in indoor environments: Geometric, topological, and semantic navigation," in *Applications of Mobile Robots*. London, U.K.: IntechOpen, Nov. 2018, doi: 10.5772/INTECHOPEN.79842.

[4] B. X. Chen, R. Sahdev, and J. K. Tsotsos, "Integrating stereo vision with a CNN tracker for a person-following robot," in *Computer Vision Systems* (Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10528. Berlin, Germany: Springer-Verlag, 2017, pp. 300–313.

[5] M. Wang, D. Su, L. Shi, Y. Liu, and J. V. Miro, "Real-time 3D human tracking for mobile robots with multisensors," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5081–5087, doi: 10.1109/ICRA.2017.7989593.

[6] W. P. Chan, S. Radmard, Z. Q. Hew, J. Morris, E. Croft, and H. F. M. Van der Loos, "Autonomous person-specific following robot," Oct. 2020, *arXiv:2010.08017*. Accessed: Jul. 9, 2021.

[7] I. Condés, J. M. Cañas, and E. Perdices, "Embedded deep learning solution for person identification and following with a robot," in *Proc. Workshop Phys. Agents*, in Advances in Intelligent Systems and Computing, vol. 1285, Nov. 2021, pp. 291–304, doi: 10.1007/978-3-030-62579-5_20.

[8] H. Liu, J. Luo, P. Wu, S. Xie, and H. Li, "People detection and tracking using RGB-D cameras for mobile robots," *Int. J. Adv. Robot. Syst.*, vol. 13, no. 5, Sep. 2016, Art. no. 172988141665774, doi: 10.1177/1729881416657746.

[9] A. Nguyen and Q. Tran, "Autonomous navigation with mobile robots using deep learning and the robot operating system," Dec. 2020, *arXiv:2012.02417*. Accessed: Jun. 18, 2021.

[10] TensorFlow. (2021). *Models/Research/Object_Detection at Master? TensorFlow/Models*. Accessed: Jul. 26, 2022. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/object_detection

[11] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6568–6577.

[12] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9905, Dec. 2015, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[14] X. Lu, X. Kang, S. Nishide, and F. Ren, "Object detection based on SSD-ResNet," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Dec. 2019, pp. 89–92, doi: 10.1109/CCIS48116.2019.9073753.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[16] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, p. 279, Jan. 2021, doi: 10.3390/ELECTRONICS10030279.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 2961–2969, Feb. 2017.

[18] J. Luiten, P. Voigtlaender, and B. Leibe, "PReMVOS: Proposal-generation, refinement and merging for video object segmentation," in *Proc. Asian Conf. Comput. Vis.*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 11364, Jul. 2018, pp. 565–580.

[19] J. Luiten, P. Torr, and B. Leibe, "Video instance segmentation 2019: A winning approach for combined detection, segmentation, classification and tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 709–712, doi: 10.1109/ICCVW.2019.00088.

[20] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[21] L. Yang, Y. Fan, and N. Xu, "Video instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5187–5196, doi: 10.1109/ICCV.2019.00529.

[22] C.-C. Lin, Y. Hung, R. Feris, and L. He, "Video instance segmentation tracking with a modified VAE architecture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13144–13154, doi: 10.1109/CVPR42600.2020.01316.

[23] A. Ahmadi, L. Nardi, N. Chebrolu, and C. Stachniss, "Visual servoing-based navigation for monitoring row-crop fields," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 4920–4926, doi: 10.1109/ICRA40945.2020.9197114.

[24] Y. Li and J. Kosecka, "Learning view and target invariant visual servoing for navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 658–664, doi: 10.1109/ICRA40945.2020.9197136.

[25] (Mar. 2, 2022). *ROS Documentation. Quaternion Message*. Accessed: May 16, 2022. [Online]. Available: http://docs.ros.org/en/noetic/api/geometry_msgs/html/msg/Quaternion.html

[26] C. Gohlke, "Homogeneous transformation matrices and quaternions," Lab. Fluorescence Dyn., Univ. California, Irvine, CA, USA, Tech. Rep., Release v2021.6.6, Jun. 2021. Accessed: May 16, 2022. [Online]. Available: https://github.com/cgohlke/transformations/

[27] W. Abdulla. (2017). *Mask R-CNN for Object Detection and Instance Segmentation on Keras and TensorFlow*. Accessed: May 15, 2022. [Online]. Available: https://github.com/matterport/Mask_RCNN

[28] W. Abdulla. (2018). *Release Mask R-CNN 2.1*. Accessed: May 15, 2022. [Online]. Available: https://github.com/matterport/Mask_RCNN/releases/tag/v2.1

[29] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, "Convolutional neural network (CNN) for image detection and recognition," in *Proc. 1st Int. Conf. Secure Cyber Comput. Commun. (ICSCCC)*, Dec. 2018, pp. 278–282, doi: 10.1109/ICSCCC.2018.8703316.

[30] B. Alsallakh, N. Kokhlikyan, V. Miglani, J. Yuan, and O. Reblitz-Richardson, "Mind the pad—CNNs can develop blind spots," Oct. 2020, *arXiv:2010.02178*.

[31] The TensorFlow Authors. (2015). *Primitive Neural Net (NN) Operations*. Accessed: Sep. 10, 2022. [Online]. Available: https://github.com/tensorflow/tensorflow/blob/v2.10.0/tensorflow/python/ops/nn_ops.py

[32] D. E. Joyce. (1997). Laws of cosines & sines. Clark University. Accessed: Oct. 26, 2022. [Online]. Available: https://www2.clarku.edu/faculty/djoyce/trig/laws.html

[33] D. Veljan, "The 2500-year-old Pythagorean theorem," *Math. Mag.*, vol. 73, no. 4, pp. 259–272, 2000, doi: 10.1080/0025570X.2000.11996853.

[34] TurtleBot. (2021). *TurtleBot/TurtleBot_Description at Melodic? TurtleBot/TurtleBot*. Accessed: Oct. 28, 2022. [Online]. Available: https://github.com/turtlebot/turtlebot/tree/melodic/turtlebot_description

[35] S. S. Sumit, J. Watada, A. Roy, and D. Rambli, "In object detection deep learning methods, YOLO shows supremum to mask R-CNN," *J. Phys., Conf. Ser.*, vol. 1529, no. 4, Apr. 2020, Art. no. 042086, doi: 10.1088/1742-6596/1529/4/042086.

[36] Ultralytics. (2023). *YOLOv8*. Accessed: Apr. 1, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[37] A. Dominguez-Sanchez, M. Cazorla, and S. Orts-Escolano, "Pedestrian movement direction recognition using convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3540–3548, Dec. 2017, doi: 10.1109/TITS.2017.2726140.

ISAAC ASANTE (Member, IEEE) received the bachelor's degree in information and communication technology (major in software technology) and the M.Sc. degree (by research) from the Swinburne University of Technology, Kuching, Sarawak, Malaysia, in 2021 and 2023, respectively. His research interests include computer vision, deep learning, robotics, and software engineering. He is a member of the IEEE Young Professionals Community.

LAU BEE THENG (Senior Member, IEEE) is currently a Professor with ICT and the Associate Dean of research and development with the Swinburne University of Technology, Sarawak. She has been actively contributing to her research areas with various edited books, peer-reviewed journals, conference proceedings, masters by research and Ph.D. completions. She has funded research projects on assistive technologies for special children, facial expression recognition-based communication, social skills acquisition with animations, real-time behavior recognition, smart technologies for the visually impaired, creative art therapies for autism, and STEM education. In addition, she is actively involved in community services working with special education schools and NGOs for the people with visual impairment and other disabilities.

MARK TEE KIT TSUN (Member, IEEE) received the B.Sc. degree (Hons.) in computer science from Coventry University, in 2005, and the B.Eng. (Hons.) degree, master's degree in software engineering (OUM), and the Ph.D. degree from the Swinburne University of Technology, Sarawak, in 2014 and 2018, respectively. He is currently pursuing research and industrial applications of the Internet of Things (IoT), deep learning, and assistive robotics.

He joined the Faculty of Engineering, Computing and Science, Swinburne University of Technology, as a Lecturer. He has served and intermittently contracted in the software development industry until 2018. His doctoral research focused on developing a real-time multi-sensor fusion model for augmenting the human-following navigation of indoor companion robots. His previous activity areas include computer game development, drone technology applications, and assistive robotics for injury prevention.

HUDYJAYA SISWOYO JO (Senior Member, IEEE) received the Bachelor of Engineering degree (majoring in robotics and mechatronics) and the Ph.D. degree from the Swinburne University of Technology, Sarawak. He was a Teaching Assistant with Swinburne University of Technology, while continuing his doctorate research, from 2008 to 2012. In 2012, he joined Swinburne University of Technology, as a Lecturer with the Faculty of Engineering, Computing and Science, where he was appointed as a Senior Lecturer, in 2018. He has been actively engaged in research related to robotics and mechatronics, automation and mechanization, and supervising students in various projects and international/national level competitions. He has completed several manufacturing and automation projects focusing on Industry 4.0 implementation for small and medium-sized enterprises in Sarawak. His research interests include mechatronics system design, modeling and control, industrial automation, human–machine interaction, and agriculture mechanization.

CHRIS MCCARTHY received the Ph.D. degree from Australian National University, in 2010. He is an Associate Professor with the Department of Computing Technologies, School of Science, Computing and Engineering Technologies (SoSCET), Swinburne University of Technology, Hawthorn, VIC, Australia. He is a Stream Leader with the Swinburne Smart Cities Research Institute, where he has been leading the Intelligent Transport Systems and Infrastructure Stream. He also serves as an Academic Director (WIL) with SoSCET. During his Ph.D., he worked on novel biologically-inspired computer vision for robot navigation (based on honeybee vision). He also spent six months as a Visiting Researcher with the Italian Institute of Technology's Robotics, Brain and Cognitive Sciences Group, Genoa, in 2007, where he worked on optical flow-based threat detection algorithms as part of the EU-funded iCub humanoid robot project. Over his academic career, he has been formally recognized for teaching, research, and industry engagement. His research interests include computer vision algorithms for robotics, intelligent transport systems, and human–computer interaction (for assistive technologies).