Real-time GP-based Wheelchair Corridor Following

Ammar TELLO-1 and A. H. Abdul HAFEZ-2 and Bakr SARAKBI-3

Computer Engineering
Hasan Kalyoncu University
Gaziantep, Turkey
ammar.tello@std.hku.edu.tr-1, abdul.hafez@hku.edu.tr-2, bakr.sarakbi@hku.edu.tr-3

***Abstract*—In this paper, we present a novel GP-based visual controller. The HOG features are used as a global representation of the observed image. The Gaussian Processes (GP) algorithm is trained to learn the mapping from the HOG feature vector onto the velocity variables. The GP training is achieved using corridor images collected from different places, these images are labeled using velocity values generated by a geometric-based control law and robust features. A hand-based verification of the features is done to ensure the accuracy of the ground truth labels.Experiments were conducted to explore the capabilities of the developed approach. Results have shown R Squared metric with more than ninety percent on the trained GP model in noisy conditions.**

***Keywords*—Gaussian Processes, Visual Servoing, Wheelchair.**

2021 29th Signal Processing and Communications Applications Conference (SIU) | 978-1-6654-3649-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/SIU53274.2021.9478010

I. Introduction

The corridor following task was achieved in [1] as a visual servoing task. It presented as a straightforward application of the Image-based visual servoing. The aim is to navigate through corridors autonomously, where people with multiple disabilities find themselves in a challenging situation [2] [3]. Another attempt has been reported in [4] to accomplish the same task. They employ an RGB-D camera in addition to a bunch of other sensors which add more cost into the system.

The corridor following task was also implemented in [5] using a learning-based method where a ResNet-18 CNN model is trained on the ImageNet dataset and fine-tuned on a corridor specified dataset. This approach has outperformed the traditional one in terms of the robustness of the extracted features. However, a heavy training phase is needed. In addition, the inference of such a deep model on a low cost embedded board in terms of time and computational complexity is not applicable. This learning-based method can be categorized under the Direct Visual Servoing (DVS) scheme where the whole image is considered as an input to produce the desired control signal [1]. The implementation of a robotic application such as corridor following on the base of DVS hard to be scalable over the different shapes of corridors and different internal environmental conditions.

In this work, we attempt to gather the strengths of both: learning and DVS techniques while avoiding their drawbacks. The learning-based approach has proved to generalize well and to be robust but at the cost of time and computation complexity. On the other hand, the DVS was proved to be a lightweight approach but at the cost of scalability. This paper aims to develop an overarching framework that is robust, lightweight, and generalized over the desired task. In order to achieve this goal, we replace the control law (velocity estimation) component in the main DVS diagram with the Gaussian Processes (GP) learning algorithm where the input data are the HOG features. Histogram of Oriented Gradients (HOG) [6]. Such features have the advantage of being invariant to illumination changes because of the oriented gradients which make it suitable for a variety of robotic applications.

978-1-6654-3649-6/21/$31.00 ©2021 IEEE

The complete processing starting from image acquisition to producing the velocity signals were deployed into a Raspberry PI pocket computer. The execution time meets the real time requirements of the wheelchair control task, it has shown a frequency of $7Hz$.

The main contributions of this paper is a novel real time GP-based DVS framework for robotics applications. A wheelchair corridor following task is formulated as HOG-based controller and realized using a GP-based controller. The remaining of paper reviews the wheelchair model and the corridor following tasks in Section II. Section III represents the proposed approach in detail followed by the results from the conducted experiments illustrated in Section IV.

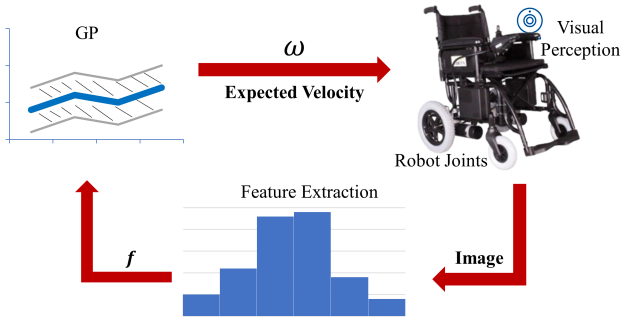
II. Wheelchair Modeling for Corridor Following

We assume that our wheelchair is a wheeled robot with six wheels, it is moving on a horizontal plate. It is actuated using the two large rear wheels by a differential controller and two DC motors. Additional two passive front wheels and two small wheels behind the wheelchair are also used to support the wheelchair. In fact, a wheelchair can be modeled as a unicycle robot [7] [8] thus matching non-holonomic constraints [1], because it has two driven wheels which are controlled independently by two direct current motors and four caster wheels to maintain stability [8]. The two control variables related to the wheelchair are then the translation velocity υ along its forward/backward direction and the angular (steering) velocity ω that perform the wheelchair rotation and direction changing.

In the global coordinate frame, the velocity of the unicycle robot where the control inputs $u = (\upsilon, \omega)$ is represented as follows [7]

$$\begin{aligned} \upsilon_x &= \upsilon \cos\varphi \\ \upsilon_y &= \omega \sin\varphi \\ \omega_z &= \omega \end{aligned} \tag{1}$$

These three equations describe the kinematic model of the robot with respect to the global coordinate frame.



Şekil 1: Block diagram for the introduced GP-based visual control framework

In the corridor task, only the angular velocity ω around axis y is considered to be calculated through the control law, while The translational velocity v^* is considered as a constant value.

III. GP-BASED FOR CORRIDOR FOLLOWING TASK

The proposed approach for corridor following task is detailed through this section. As depicted on Figure 1, the input of our framework is the current image that acquired from the attached camera, a HOG descriptor for this image is calculated and fed into the trained GP model which gives an appropriate angular velocity ω as an output of the framework.

The HOG descriptor has achieved the best performance among other global descriptors including direct image intensities and deep features. As a result, HOG was selected to be used as a global descriptor for the collected images in this work.

A. GP-based regressor as a wheelchair visual controller

Since the GP is considered a non-parametric algorithm, then the only optimization required is related to the added noise or the hyperparameters that could exist in the used kernel. To accomplish such an optimization, an optimization algorithm can be applied on the loss function $\log p(y|\theta_{hyp})$ which is the log-likelihood of the probability of witnessing our expectations conditional on the hyperparameters θ_{hyp}.

In order to get the expected value for new input, the matrix $K(X, x^*)$ consists of the kernel values between all inputs represented in the X matrix as row vectors and the new input x^*. The kernel matrix is transposed and multiplied by the inverse of the C matrix which is the noised kernel matrix.

$$m(X, x^*) = K(X, x^*)^T C^{-1} y \tag{2}$$

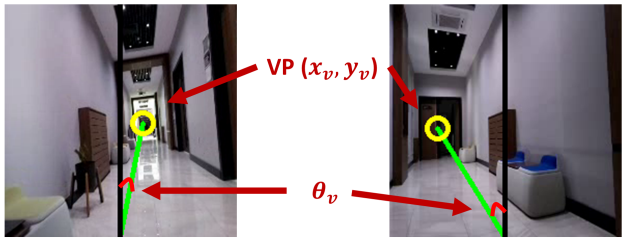
In this work, we defined the kernel k between two inputs x_i and x_j as follows:

$$k(x_i, x_j) = \exp(\frac{-1}{2l^2}(x_i - x_j)^T(x_i - x_j)) \tag{3}$$

and C is expressed as:

$$C = K(X, X) + I\sigma \tag{4}$$

It is worth noting that l and σ are considered as hyperparameters where $\theta_{hyp} = l, \sigma$, furthermore, they are optimized in the training phase of the GP through SLSQP algorithm which stands for sequential least square programming. The SLSQP was implemented as part of the Scikit-learn Python library. Regrading the initialization values of the hyperparameters, we noticed that changing the initial values does not affect the convergence process of the optimization function in terms of finding the optimum solution. It only affects the number of iterations needed to reach the optimum solution.



Şekil 2: Examples of the features detected by a human annotator to be used as a ground truth where VP is the vanishing point with coordinates (x_v, y_v) and θ_v is the angle between the median line and middle of the image.

We are interested in calculating the angular velocity ω around axis y only. Hence, the GP-based control law can be derived as a single output GP which is shown as follows:

$$\omega_y = GP_{\omega_y}(HOG) \tag{5}$$

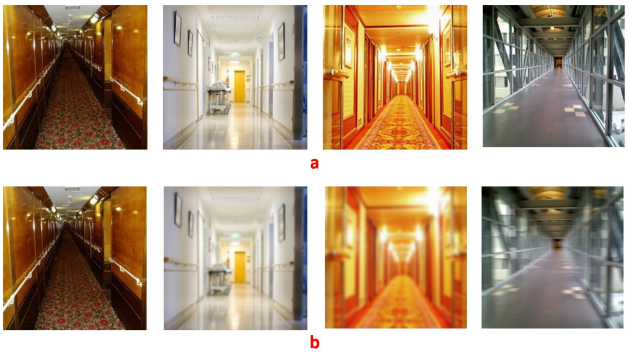
It is clear that GP can calculate the desired velocity based only on HOG features with no need to know the reference features in prior in order to calculate the error function as in the traditional visual servoing techniques.

B. Training data and labeling

Calculating the ground truth of the gathered data is an essential part in order to train a proper GP model for the specified task. The ground truth (angular velocities) of the dataset was calculated through the traditional approach introduced in [1], whereas the vanishing point and the median line were detected by a human annotator to gather more accurate coordinates of those geometric features and some examples are given in Figure 2. After the detection step, the angular velocity is calculated for each input image by the control law proposed in [1]. The ground truth for each image is calculated and inserted into a one vector y.

Our dataset contains images of corridors from different viewpoints and also different architectural features which gathered using the open-access datasets published by [9] and [10] in addition to the images collected by [5].

The gathered data consisted of 2610 images. Those images were duplicated and randomly noised with one of four different noises which are as follows: JPEG Compression, Motion Blur, Mild and Strong Gaussian. To accomplish this task, we looped over the 2610 images, and for each image, we picked one of the four available noise types randomly to be applied to the current image. The JPEG compression was one of the noise types because the images are in PNG format which is not the normal case because of the large storage space it requires, so, this compression can simulate a more realistic situation.



Şekil 3: Some examples from the collected dataset (a) Four images from the Clean subset (b) Four images from the noisy subset corresponding to images in the first row where the kind of noise that applied from left to right is: JPEG Compression, Mild Gaussian, Strong Gaussian and Motion Blur.

Then, we have the Motion Blur which simulates the blur in images caused by the fast robot movement. Finally, the Mild and Strong Gaussian noises are to show how our model will react to different levels of blur caused by inappropriate changes in the focus of the camera.

As a result, the whole dataset become equal to 5220 images. Some examples are shown on Figure 3.

IV. Experimental Evaluation

This section demonstrate the generalization and robustness of our approach. In the next Subsection we show the robustness and error analysis along with analysis matrices. After that, our real wheelchair experiments are presented.

A. Analysis metrics

MSE is used in this work and it is defined as follows:

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(\omega_k - \widehat{\omega_k})^2 \tag{6}$$

where ω represents the ground truth of angular velocities, $\widehat{\omega}$ represents the predicted velocities and n is the number of the test cases. A lower MSE value means a lower error and it is not restricted into a specific range.

R^2 was also used to measure the robustness of our model as given below:

$$R^2(\omega, \widehat{\omega}) = 1 - \sum_{k=1}^{n}\frac{(\omega_k - \widehat{\omega}_k)^2}{(\omega_k - \bar{\omega}_k)^2} \tag{7}$$

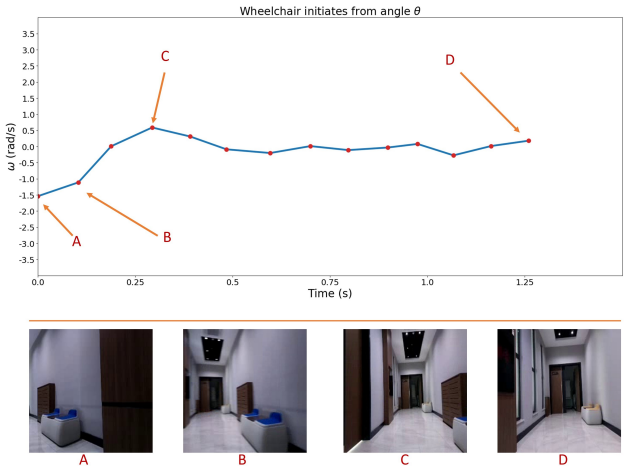
where $\bar{\omega}$ is the mean of the ground truth values related to the test images. The values of R^2 start from $-\inf$ into 1 where 1 is the best value.

B. Analysis of The Performance With Clean and Noisy Images

In this experiment, the GP training and testing was conducted first on the "Clean" dataset in order to observe the performance of the proposed control law in the ideal state. And then it followed by training and testing on the whole dataset i.e. clean and noisy images to analyze the performance of the GP-based DVS model on the different noises types. HOG parameters for the both situations were set into 32×32 pixel per cell represented by 9 bins (features) for each. HOG parameters were empirically set to give the highest performance. The datasets was divided into train and test with percentage of 90% and 10% respectively. For the clean dataset, the MSE gives 0.2332 where the ground truth values after normalization has a margin which approximately equal to 12. As a result the error is equal to 1.94% which a very small percentage. R^2 gives a percentage equal to 75.34%. This result need to be better, and this is achieved through generalizing the trained GP model to be trained and tested on a more realistic dataset as shown in the second part of this experiment.

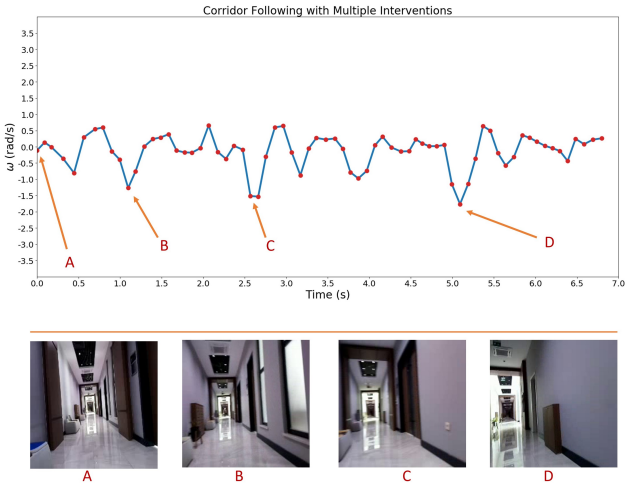
Test Set	MSE	R^2 GP-based	R^2 CNN-based
JPEG Compression	0.0133	98.52	88.57
Mild Gaussian	0.0428	94.19	88.34
Strong Gaussian	0.1501	83.74	79.97
Motion Blur	0.1355	86.83	88.32
Clean	0.0836	91.02	88.32
Full	0.0983	90.70	

TABLO I: R^2 for HOG tested on six different sets



Şekil 4: The upper figure shows a curve of the angular velocities ω along with turn movement of the wheelchair movement. The images with names A, B, C and D show the camera view captured at the corresponding positions mentioned in the first figure.

For the full dataset, the MSE and R^2 results are depicted on table I which represents the performance over different sets including noisy and clean in addition to a test over images from the full dataset. It is obvious that error was decreased and the performance increased a lot in comparison with the previous situation. The results are comparable for the different test sets and still gain high performance even with strong noise types like the strong Gaussian and motion blur, where the error is about 1.2% in comparison with the ground truth range which still a very tiny percentage. These results lead to the fact that the proposed model is immune to the different noise types. Furthermore, a comparison with the previous work [5] was made, as depicted on the table I, Despite that our approach has almost the same performance of the CNN-based approach for the motion blur subset, it has outperformed it for all other subsets.



Şekil 5: The upper figure shows a curve of the angular velocities ω along the wheelchair movement. The images with names A, B, C and D show the camera view captured at the corresponding positions mentioned in the first figure.

C. Experiments with real Wheelchair Setup

A real wheelchair experiments were conducted for the developed visual GP-based controller to achieve the corridor following task. An electric wheelchair from "Volta" was used. A single Raspberry PI equipped with a camera module is used for the complete computations starting from images acquisition to velocity calculation. No other computer or computation device are used where our trained GP model was efficiently deployed on the Raspberry PI. The angular velocity is sent to the "Sabertooth" motor driver in order to differentially drive the wheelchair motors. The method works on about $7Hz$ frequency, corresponding to $143ms$ for executing the full motion cycle.

1) Start from arbitrary position: The aim of this experiment is to prove that our trained model can align and drive the wheelchair through the corridor starting from an arbitrary position, not aligned with the corridor. In this experiment, large part of the main structure of the corridor was initially out of the camera view field. The wheelchair was able to correct its direction and align with the corridor. As depicted in Figure 4, the first shot marked with (A) letter shows the image captured by the camera at the initial moment. The negative value of the corresponding angular velocity at the position (A) works on correcting the direction of the wheelchair. The correction takes several time moments, the progress of it is shown in the next images i.e. B, C, and D. It is clear that the wheelchair has become in alignment with the corridor at the position of image C.

2) Corridor Following with Multiple Interventions: In this experiment, we measure the robustness of our model in terms of the response to unexpected interventions that could lead to an inappropriate response while navigating through the corridor. Interventions were made along the corridor by pushing the wheelchair towards the wall. In Figure 5, the first image (A) show the start position. Images B, C, and D represent the main interventions, where each image shows the camera view at the direction which caused by each intervention. The curve shows the corresponding velocity response to each of the mentioned interventions. This experiment demonstrates that after every interventions made through the corridor, our model is still able to correct the error and get the wheelchair back into the right path.

V. CONCLUSIONS

This proposed approach is applied to the corridor following task by an electric wheelchair. Furthermore, the robustness of our approach against different challenges is demonstrated as well. The approach utilises a GP-based visual controller. The GP considers the global HOG features as an input and was trained to produce the velocity signal as an output. The advantages of the GP algorithm was exploited in terms of the lightweight, performance, and noise immunity. Despite the fact that our approach was trained over a small number of images, the conducted experiments proved the robustness and real time performance of our approach in different real-world scenarios. In the future works, we aim at more scalable and trainable features.

ACKNOWLEDGMENT

This work is partially supported by TUBITAK under project number 117E173 and by the CARA Syria program. The Titan Xp used for this research was donated by the NVIDIA Corporation.

REFERENCES

[1] F. Pasteau, V. K. Narayanan, M. Babel, and F. Chaumette, "A visual servoing approach for autonomous corridor following and doorway passing in a wheelchair," *Robotics and Autonomous Systems*, vol. 75, pp. 28–40, 2016.

[2] A. Murarka, S. Gulati, P. Beeson, and B. Kuipers, "Towards a safe, low-cost, intelligent wheelchair," in *Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV)*. Citeseer, 2009, pp. 42–50.

[3] P. Geoerg, J. Schumann, S. Holl, and A. Hofmann, "The influence of wheelchair users on movement in a bottleneck and a corridor," *Journal of Advanced Transportation*, vol. 2019, 2019.

[4] M. Burhanpurkar, M. Labbé, C. Guan, F. Michaud, and J. Kelly, "Cheap or robust? the practical realization of self-driving wheelchair technology," in *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2017, pp. 1079–1086.

[5] V. S. Dorbala, A. A. Hafez, and C. Jawahar, "A deep learning approach for robust corridor following from an arbitrary pose," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2019, pp. 1–4.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 886–893.

[7] S. Gulati and B. Kuipers, "High performance control for graceful motion of an intelligent wheelchair," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3932–3938.

[8] V. H. Andaluz, P. Canseco, J. Varela, J. S. Ortiz, M. G. Pérez, V. Morales, F. Robertí, and R. Carelli, "Modeling and control of a wheelchair considering center of mass lateral displacements," in *Intelligent Robotics and Applications*. Springer, 2015, pp. 254–270.

[9] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos, "Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets," in *In proceedings of IROS*, vol. 6, 2006, p. 93.

[10] S. Yang, D. Maturana, and S. Scherer, "Real-time 3d scene layout from a single image using convolutional neural networks," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 2183–2189.