Trends and choices in prototyping vision-based control

Vlad Ovidiu Piacentini Mihalca[1] **Tocut Pavel Danuț**[1]

1 University of Oradea, Faculty of Management and Technological Engineering,

ovidiu@vmihalca.ro

Abstract: This paper analyzes a vision-based control system for mobile robots that performs target tracking using image-based visual servoing. The system employs a distributed architecture in which vision processing is executed on a central workstation, while mobile robots act as sensor platforms and end-effectors, enabling rapid prototyping and the use of computationally demanding algorithms. Visual features are extracted either through classical methods or Deep Learning–based segmentation, while control is handled by an explicit software controller. Using a modern taxonomy, the system is classified as an Explicit Control Structure and Single-Target Tracking Visual Servoing scheme. Although Deep Learning components are incorporated, the approach does not constitute an end-to-end solution, as the control law remains separate from the learning model. The paper discusses design trade-offs and outlines future directions toward end-to-end, AI-based visual control methods.

1 Introduction

Vision-based control can be described as mimicking natural systems in Robotics through the use of input information of a visual nature, coming from camera sensors, to control end-effectors towards a specific goal or task. End effectors can be industrial robotic arms or mobile robots. This approach stands in contrast to using different input data such as distance or location, which require other sensors (ultrasonic, infrared, GPS) that do not resemble biological eyes.

Until recently, it can be confidently said that visual information requires processing before being used in a decision-making step within a control scheme. This processing relies on standard Computer Vision algorithms and procedures upon raw image data, captured by the sensors. Additional processing may be used for avoiding data loss [1], [2] or it can be achieved through data validation modules [3], [4].

In the 3rd edition of his book on Vision-based control, Peter Corke makes a statement in a sub-chapter dedicated to image segmentation. Specifically, he describes that classic segmentation methods “are suitable for relatively simple scenes and have the advantage of requiring only modest amounts of computation. However, in recent times these techniques have been eclipsed by those based on deep learning, which is a powerful and practical solution to complex image segmentation problems.” [5].

This statement serves as the motivation for the current paper. A discussion on the categories and classification of modern vision-based control is proposed, followed by an analysis of a visual control scheme proposed as part of a research project. The project had the goal of studying vision-based control by means of new systems, control scheme implementations and simulations of robot kinematic models, continuing the work from [6].

Within the research project mentioned a prototyping platform was developed, comprised of a central workstation and a group of mobile robots[7]. The system was inspired by educational projects such as experiments that can be carried remotely in a lab[8] or Augmented Reality applications with remote control[9].

2 A classification of modern vision-based control

In their 2022 article Machkour, Ortiz-Arroyo and Durdevic propose a classification of Visual Servoing methods into several categories that better encompass the range of methods found in practice.[10] They split the methods into classic and new approaches, for the latter proposing a taxonomy:

Classic Visual Servoing

1. Image-based (IBVS)
2. Position-based (PBVS)
3. Hybrid

Modern Visual Servoing

1. Direct visual servoing
2. End-to-end systems
3. Fixed/ moving Target Tracking
4. Single/ multiple Target Tracking
5. Explicit/ implicit controller

In classic Visual Servoing, image-based methods define an error function that operates on image coordinates for visual features. These features are extracted with various image segmentation algorithms. Controllers are defined by directly relating change in feature set s coordinates to velocities vector ν, which becomes the output. Such a relation is mediated by a feature sensitivity matrix Ls also known as interaction matrix or image Jacobian.

$$\dot{s} = \mathbf{L}_s \cdot v \tag{2.1}$$

Position-based methods introduce an additional step for pose estimation. This is achieved through various Computer Vision algorithms, such as Perspective-n-Point (PnP) which represents a foundational problem of determining a camera's pose through mapping 3 tridimensional points to their projections in 2D. After the pose estimation step, the error function is defined in 3D space on target and camera pose, controllers relying on spatial math to modify the pose of the robot.

Hybrid Visual Servoing methods have error functions which include both image features and 3D space coordinates. An example definition of such an error function e can be found in [11].

Among the new methods, direct Visual Servoing does not make use of features. It considers all the image data as the feature set.

$$s(\xi) = I_\xi \tag{2.2}$$

The method is based on the principle of computing a similarity measure between images and adopting a suitable function becomes an important aspect of the implementation. Such a method, named photometric Visual Servoing is explained in-depth within [12]. Choosing a feature set, together with all the image processing involved for it, can be avoided by defining the feature set as the luminance of all image pixels. The error function can then be defined as a difference between current image and a target image:

$$e = \mathbf{I} - \mathbf{I}^* \tag{2.3}$$

The paper provides further details regarding the computation of an interaction matrix based on the luminance I for all the image points.

In end-to-end methods there are no controller components involved. Instead, the control law is embedded into Deep Learning models or learned through Machine Learning techniques. The error function can be defined as a mean error between the process value and model inference output (predictions).

Given that target tracking is an idiomatic task for Vision-based control applications, the authors of [10] include several categories of Visual Servoing methods related to this task. Regarding the motion of the target, there are the Fixed Target Tracking Visual Servoing (FXTTVS) and the Mobile Target Tracking Visual Servoing (MBTTVS) categories. For a moving target its dynamics need to be estimated, therefore the feature set includes elements related to target dynamics.

With regards to the number of targets involved, two specific categories were proposed: Single Target Tracking Visual Servoing (STTVS) and Multiple Target Tracking Visual Servoing (MTTVS). In STTVS there is only one target which belongs to a class, while MTTVS includes multiple targets all belonging to the same class.

The final categories, explicit (ECS) or implicit controller Visual Servoing (ICS), denote whether the system contains a feedback control loop component. Classic Visual Servoing methods belong to the explicit category, as well as DL-based methods which have a controller module. In the case of implicit methods category, the control law is learned by means of embedding it inside a DL model.

3 Architecture considerations for prototyping systems

The system described in [13] implements vision-based control by means of defining features and extracting them from the image with a D.L.-based component. It achieves a target tracking task for a mobile robot.

As described in [7] the system closes a remote control loop as it is distributed over mobile robot agents and a central workstation for carrying out Vision computations and algorithms. This allows for fast prototyping vision-based control schemes as it eliminates deployment steps onto the robots. The mobile robots act as information gatherers and are also end-effectors in the system, receiving remote commands. This structure also benefits from increased processing power and allows for implementing more demanding algorithms, while having an increased tolerance to un-optimized or naïve code.

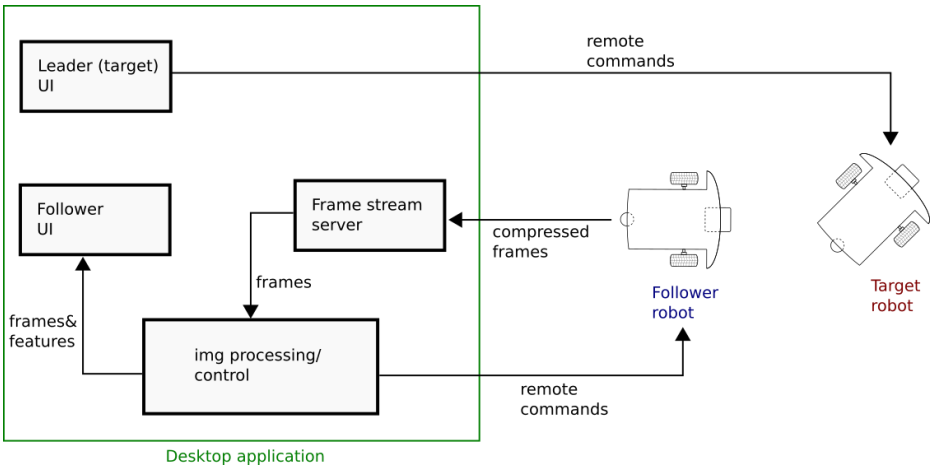

Figure 1 System architecture diagram[7]

As can be seen in Figure 1, the loop is closed over the network. This particular implementation makes use of two wireless TCP connections: one for sensor data flow towards the central workstation and one for remote control of the mobile robots. The data flow is implemented as a simple stream of information, while remote control relies on existing RPC solutions.

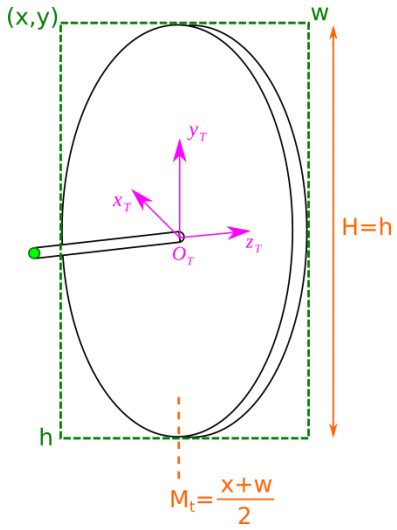

Figure 2 Target visual features[14]

The visual control scheme itself relies on features defined in a mathematical model of the system, which are extracted from the image (Figure 2). It is a classic Visual Servoing method and it is image-based as it operates on feature set coordinates.

Considering the previously defined taxonomy, the system can be classified as belonging to the ECS category, as it contains an explicit software controller module. There are multiple versions of the scheme that were prototyped and studied: based on classic feature extraction as well as Deep Learning-based solutions. It is important to mention that in all cases, the controller component still exists as part of the system.

The scheme can also be classified as STTVS, as it is specialized in achieving a target tracking task using the remote-controlled mobile robots group. There is only one target detected by the segmentation procedure, belonging to a single class of objects.

Despite the fact that a version of the visual control scheme contains a Deep Learning component, it cannot be classified as an end-to-end system. The model is used only for image segmentation, feature extraction is still present in that specific version. The control law is not embedded into the Deep Learning component.

The component makes use of the Transfer Learning property, which allows a pre-trained model to be used as the base for a specialized model, the specialization occurring in the final layers towards the output (Figure 3). This final layer acts as the regressor for bounding box coordinates, achieving target object detection. This idea is inspired from the paper by Komorowski et al. which detect similar circular-shaped targets, making use of a pyramidal architecture[15]. The FootAndBall model presented by the authors propagate features from a series of convolutional layers to a superior dense layer which acts as a regressor in a similar way to the model proposed by our visual control scheme. Such a model architecture implementation for a detector is discussed in several web resources[16], [17].

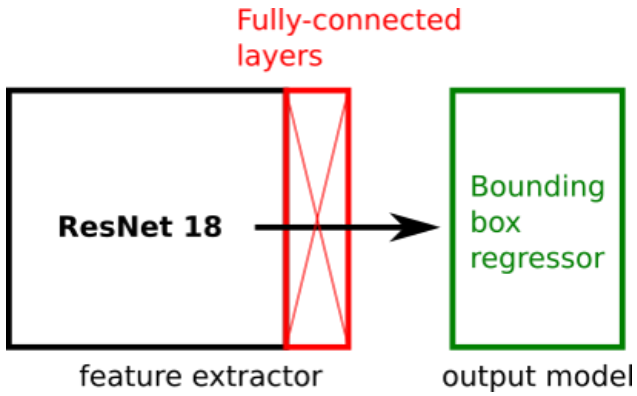

Figure 3 D.L. detector layers[13]

Object detection represents a classic problem for Computer Vision and there are other well-established approaches to it: among state-of-the-art detectors are mentioned YOLO and SSD which are efficient and achieve the detection task in a single pass[17], [18].

Re-stating the problem to fit an end-to-end system implies embedding the control law into the Deep Learning model. Therefore, the component is not isolated to an object detection sub-task, but rather its output changes to the velocities required to minimize the error function value (Figure 4).

Figure 4 End-to-end component block diagram

4 Closing remarks

In the current state of technology, end-to-end solutions have a promising potential for accuracy at the cost of hardware resources. A benefit of segmentation approaches and explicit controllers lie in the possibility for simple and optimized implementations which can run on robotics hardware without the need for dedicated computational modules or specialized SoC. Such simple approaches rely on assumptions and have an increased risk of reduced accuracy when these assumptions are not met.

The visual control scheme analysed by the authors of this paper has been classified according to a modern taxonomy into several overlapping categories, sketching a discussion on strengths and weaknesses for each design decisions regarding its core components.

This paper paves the way for future implementations and a thorough comparison of modern vision-based control approaches compared to the classic methods employed. Future directions include a greater reliance on Deep Learning for embedding control laws and a shift towards Artificial Intelligence based methods.

References:

[1] D. Noje, I. Dzitac, N. Pop, and R. C. Țarcă, “IoT Devices Signals Processing Based on Shepard Local Approximation Operators Defined in Riesz MV-Algebras,” Inform. Neth., vol. 31, no. 1, pp. 131–142, 2020, doi: 10.15388/20-INFOR395.

[2] D. Noje, R. C. Țarcă, I. Dzitac, and N. Pop, “IoT Devices Signals Processing based on Multi-dimensional Shepard Local Approximation Operators in Riesz MV-algebras,” Int. J. Comput. Commun. Control, vol. 14, no. 1, pp. 56–62, 2019.

[3] D. Noje, R. C. Țarcă, N. Pop, A. O. Moldovan, and O. G. Moldovan, “Automatic System Based on Riesz MV-algebras, for Predictive Maintenance of Bearings of Industrial Equipment Using Temperature Sensors,” in ICCCC 2022: Intelligent Methods Systems and Applications in Computing, Communications and Control, Springer, Cham, 2022, pp. 3–19. doi: 10.1007/978-3-031-16684-6_1.

[4] D. C. Noje, O. G. Moldovan, D. I. Țarcă and R. C. Țarcă, ”Image Processing Using Shepard Local Approximation Operators Defined in Riesz MV-Algebras”. In 2024 10th International Conference on Mechatronics and Robotics Engineering (ICMRE), 2024, pp. 260-264, IEEE, doi: 10.1109/ICMRE60776.2024.10532201.

[5] P. Corke, Robotics, Vision and Control - Fundamental algorithms in Python, 3rd ed., vol. 146. in Springer Tracts in Advanced Robotics, vol. 146. Springer International Publishing AG, 2023. doi: 10.1007/978-3-031-06469-2.

[6] R. C. Țarcă, Conducerea roboților utilizând sisteme servovizuale. Oradea: Editura Universității din Oradea, 2001.

[7] V. O. Mihalca, D. M. Anton, and R. C. Țarcă, “Prototyping platform design for visual robot control experiments,” Ann. Univ. Oradea Facsicle Manag. Technol. Eng., no. 1, p. 1, 2023.

[8] R. C. Țarcă, “Virtual and Remote Control Lab Experiment Using Matlab,” Ann. ORADEA Univ. Fascicle Manag. Technol. Eng., vol. XIX (IX), no. 3, pp. 78–81, 2010, doi: 10.15660/auofmte.2010-3.2017.

[9] R. Tarca, L. Csokmai, T. Vesselenyi, I. Tarca and F.P. Vladicescu, “Augmented Reality Used to Control a Robot System via Internet”. In Technological Developments in Education and Automation, pp. 539-544. Dordrecht: Springer Netherlands, https://doi.org/10.1007/978-90-481-3656-8_98.

[10] Z. Machkour, D. Ortiz-Arroyo, and P. Durdevic, “Classical and Deep Learning based Visual Servoing Systems: a Survey on State of the Art,” J. Intell. Robot. Syst. Theory Appl., vol. 104, no. 1, 2022, doi: 10.1007/s10846-021-01540-w.

[11] E. Malis, F. Chaumette, and S. Boudet, “2D 1/2 Visual Servoing,” INRIA, 2006. [Online]. Available: https://hal.inria.fr/inria-00073302

[12] C. Collewet, E. Marchand, C. Collewet, and E. Marchand, “Photometric visual servoing,” INRIA, 2008.

[13] V. O. Mihalca, O. G. Moldovan, I. Țarcă, D. M. Anton, and D. Noje, “Integrating deep learning in target tracking applications, as enabler of control systems,” in 2024 10th International Conference on Computers Communications and Control (ICCCC), Oradea, 2024.

[14] V. O. Mihalca, “Artificial intelligence aspects of interactions between agents of mobile robot systems,” University of Oradea, Oradea, 2024.

[15] J. Komorowski, G. Kurzejamski, and G. Sarwas, “Footandball: Integrated player and ball detector,” VISIGRAPP 2020 - Proc. 15th Int. Jt. Conf. Comput. Vis. Imaging Comput. Graph. Theory Appl., vol. 5, pp. 47–56, 2020, doi: 10.5220/0008916000470056.

[16] S. Chilamkurthy, “Transfer Learning for Computer Vision Tutorial.” Accessed: Jun. 24, 2024. [Online]. Available: https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

[17] D. Chakraborty, “Training an object detector from scratch in PyTorch.” Accessed: Jun. 24, 2024. [Online]. Available: https://pyimagesearch.com/2021/11/01/training-an-object-detector-from-scratch-in-pytorch/

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[19] W. Liu et al., “SSD: Single Shot MultiBox Detector,” in Computer Vision -- ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 21–37.