2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning-Based Fusion for Robust Multi-Spectral Visual Servoing

Enrico Fiasché[1,2], Siddharth Singh Savner[2], Ezio Malis[2] and Philippe Martinet[2]

Abstract— **Multispectral sensors, which measure multiple wavelength bands beyond the standard red, green, and blue channels, capture richer information than conventional RGB cameras. Such enriched data is especially valuable in visual servoing, where robot control critically depends on image content. However, leveraging multiple spectral bands (typically around a dozen) directly within real-time visual servoing constitutes a significant challenge. The only prior work tackled this problem using a Pixel Selection strategy based on image gradients. This paper introduces a learning-based framework to enhance Multi-Spectral Visual Servoing (MSVS) by fusing data from multispectral cameras into a single, robust representation for control. An autoencoder is employed to compress multispectral inputs into a noise-attenuated 2D image, which is then used within a standard rule-based Direct Visual Servoing (DVS) scheme. Comparison experiments both with simulated data and with a real robot in complex and unstructured environments show that the proposed learning-based fusion maintains stable convergence and improves positioning accuracy under noisy conditions while preserving computational efficiency.**

I. Introduction

Visual Servoing (VS) has long been studied in robotic control, providing a framework to guide robot motion directly from image feedback [1]. VS allows robots to perform complex tasks by constantly adjusting their position to match a target image. However, this method has a major bottleneck: it relies entirely on high-quality camera feeds and solid visual features. When deployed in real-world environments, standard RGB cameras often fail. Because they capture such a limited portion of the light spectrum, even minor lighting shifts can severely disrupt their output. Multispectral cameras offer a practical workaround. By recording across several specific wavelengths, they reveal a much deeper, more nuanced view of a scene. This added spectral data exposes physical details that RGB lenses simply cannot see, which is critical for demanding jobs like outdoor navigation or vegetation monitoring. Ultimately, feeding multispectral data into a VS system creates a much more resilient setup, especially when lighting conditions are less than ideal. Bringing multispectral data into real-time VS is not without its difficulties. Every single acquisition yields a large stack of N images (typically 5-15) which must be heavily compressed to meet real-time processing demands. Making multispectral imaging viable for closed-loop VS therefore relies entirely on efficient dimensionality reduction. Traditional techniques tackle this by isolating representative bands or pulling out key features, with common tools including Principal Component Analysis [2], Sparse Nonnegative Matrix Factorization [3], and clustering-based band selection [4]. These classical methods work well enough for offline analysis. In a live system, however, they are often too computationally heavy. Furthermore, blending spectral bands in this way tends to produce abstract features that lack clear physical interpretation. Our previous work [5] took the first real step toward integrating multispectral imaging into VS. To handle the data load, they relied on a Pixel Selection strategy, compressing the multispectral data into a single 2D image based on image gradients. Feeding this fused image into a standard Direct Visual Servoing (DVS) framework noticeably improved outdoor robustness compared to standard RGB setups. While promising, the method is inherently limited: the reliance on local gradient computations makes the fused image sensitive to noise and spectral distortions, potentially degrading performance in realistic conditions.

Building on this idea, we introduce a more general hybrid approach. Instead of relying on handcrafted rules such as Pixel Selection, we use a learning-based fusion method to compress the N multispectral bands into a compact and noise-resilient representation. The emergence of neural network-based methods has already shown powerful alternatives for spectral-spatial fusion. Convolutional architectures, in particular, can learn to extract complementary spatial and spectral features directly from multispectral or hyperspectral data. By capturing complex correlations across bands, these networks produce fused representations that retain both spectral fidelity and spatial detail. Recent examples include STDFusionNet [6], which integrates infrared and visible information, and two-branch CNNs [7] for joint spectral-spatial feature extraction. Advanced models such as Multispectral/Hyperspectral FusionNet [8] and ARGS-Diff [9] further demonstrate that deep learning can generate high-quality fused images with minimal loss of detail.

Autoencoders (AEs) [10], [11] are ideal for this task because they naturally learn to compress data into a tight latent space while preserving essential features. When combined with convolutional layers, they maintain spatial structure, making them highly effective for image fusion and dimensionality reduction. For multispectral data, AEs can squeeze multi-channel inputs into a compact feature map that reduces noise and integrates cross-channel data without losing spatial resolution. However, while prior AE-based fusion methods [12]–[14] excel at offline tasks like classification and super-resolution, their potential for real-time Visual Servoing is largely untapped. Adapting these strategies for closed-loop control is challenging: the dimen-

[1]Université Côte d'Azur DS4H, Nice, France, enrico.fiasche@inria.fr
[2]Inria Centre at Université Côte d'Azur, Sophia Antipolis, France, {siddharth-singh.savner, ezio.malis, philippe.martinet}@inria.fr

sionality reduction must be highly robust, yet fast enough to meet strict real-time timing constraints. In this work, our methodology centers on a learning-based strategy that fuses bulky multispectral inputs down to a highly compact format. We designed this representation to be versatile, it can easily take the shape of a full 2D image, a collection of 2D keypoints, or even 3D features. Because of this flexibility, it drops right into a wide variety of standard Visual Servoing setups. The underlying control framework remains completely untouched. Ultimately, we get the best of both worlds: the sheer robustness of deep learning for perception, paired with the interpretable stability of classical VS control. The main contributions of this paper break down into three main areas. First, we introduce a generalized learning-driven fusion pipeline capable of generating multispectral representations tailored for diverse VS strategies. Second, we provide a fused 2D image representation that differs from existing methods in the literature, actively suppressing noise and enriching the data fed to the controller. Third, we validate the approach through both simulations and real-world experiments in complex, unstructured environments, demonstrating improved convergence and stability under noisy conditions while maintaining comparable computational efficiency to state-of-the-art methods.

II. Learning-Based Visual Servoing Frameworks

This section reviews key literature on Learning-Based Visual Servoing, highlighting both end-to-end and hybrid approaches. We also discuss the limited research on applying these methods to multispectral images.

A. Learning-Based Visual Servoing

One major shift in the field has been toward Learning-Based Visual Servoing (LB-VS). Instead of relying on traditional analytical math, these methods train end-to-end models to map perception directly to action. Researchers have approached this in a few distinct ways. Castelli et al. [15], for example, used Gaussian Mixture Models to approximate the VS function for high-frequency control. Others tackle dynamic tracking by pairing grasp detection with velocity prediction in dual-network setups [16]. When it comes to reinforcement learning, Soft Actor-Critic frameworks have proven useful for escaping local minima [17], especially when paired with hybrid sampling to speed up convergence [18]. On the supervised side, CNNs can robustly predict control commands straight from raw pixels [19]. However, despite their undeniable flexibility in complex tasks, LB-VS methods share a common bottleneck. They are notoriously data-hungry, computationally expensive to train, and essentially act as black boxes, making it tough to guarantee safety in unpredictable, real-world environments.

B. Hybrid Learning-Based Visual Servoing

A second major shift of research builds on the idea of combining learned perception with classical control, which we refer to as Hybrid Learning-Based Visual Servoing (HL-VS). HL-VS offers a compelling middle ground. This paradigm refuses to discard classical control entirely. Instead, it restricts deep learning purely to the perception phase, tasking networks with estimating scene geometry or pose, and passes that output directly to standard analytical VS laws. Researchers favor this approach because it marries the flexibility of modern neural networks with the hard stability guarantees of rule-based controllers [20].

In applied scenarios, this translates to replacing brittle geometric feature extraction with CNNs capable of regressing camera poses directly from single frames [21]. Other systems rely on networks trained via synthetic data to infer 3D object poses, which then guide the robotic manipulation [22]. To build resilience against occlusions and lighting shifts, the work [23] fine-tuned a pre-trained CNN to supply relative camera pose estimates straight to a classical control law. This strict division of labor shines in complex tasks. For instance, agricultural robots can use CNNs to navigate the visual complexity of leaf detection, leaving the precise physical grasping to a geometric controller [24]. By fusing data-driven vision with strict mathematical control, HL-VS handles unpredictable environments with remarkable accuracy.

C. Application to Multispectral Visual Servoing

Research on Multi-Spectral Visual Servoing (MSVS) remains unusual. One study proposes a Pixel Selection strategy to fuse multispectral data into a single 2D image for a standard Direct Visual Servoing scheme. Pixels are chosen based on gradient magnitude across spectral channels to retain the most informative features for control [5]. However, as shown in Fig. 1, this method has been specifically designed to work in a DVS framework and does not generalize to other VS formulations, highlighting the need for more versatile MSVS approaches.

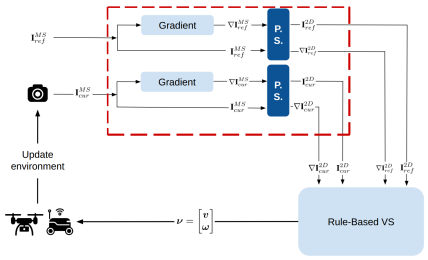


Fig. 1: Schema of the PS-MSVS framework [5].

III. Proposed Learning-Based Fusion

Extending beyond previous work in MSVS [5], we propose a Hybrid Learning-Based Multi-Spectral Visual Servoing (HL-MSVS) framework. Our framework employs a Learning-Based fusion stage that transforms the multispectral cube into a compact, noise-resilient representation, denoted by α, suitable for Visual Servoing. The fused output can be tailored to different VS formulations:

- a full 2D image for Direct Visual Servoing (DVS),
- a set of 2D keypoints for feature-based schemes,
- or 3D features for Position-Based Visual Servoing.

This flexibility allows HL-MSVS to be compatible with a broad range of VS strategies while clearly decoupling perception from control. It is also worth noting that RGB images can be regarded as a special case of multispectral data with only three channels. In conventional pipelines, RGB inputs are often reduced to grayscale through fixed linear combinations, which may limit the exploitation of spectral diversity. Alternative fusion strategies can therefore provide different representations, potentially preserving richer information for Visual Servoing.

A. Learning-Based Fusion via Autoencoders

In this work, we focus on the case where the fusion stage outputs a single 2D image for DVS. To this end, we employ autoencoders to generate compact and denoised representations that are robust to noise and suitable for real-time operation. The network learns to reconstruct an informative fused image from the multispectral cube, which is then directly integrated into a standard DVS scheme for closed-loop control, as shown in Fig. 2. This specific implementation is used primarily to enable a direct comparison with the state-of-the-art MSVS approach [5]. By replacing the handcrafted stage with a learning-based fusion, we can evaluate the benefits of HL-MSVS in terms of accuracy, robustness, and flexibility, while keeping the underlying control scheme identical.

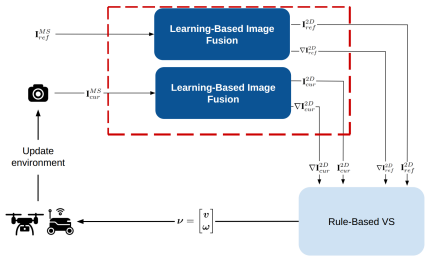


Fig. 2: Schema of the proposed Hybrid Learning-Based Multi-Spectral Visual Servoing (HL-MSVS) framework.

B. Proposed Autoencoder Architecture

We propose a generic autoencoder (AE) that learns to reconstruct clean multispectral images from noisy observations. The network follows an encoder-decoder design, as illustrated in the schematic in Fig. 3. A batch of clean multispectral images is represented by $\mathbf{I} \in [0,1]^{B\times C\times H\times W}$, where B is the batch size, C is the number of spectral channels, and (H, W) are the spatial dimensions.

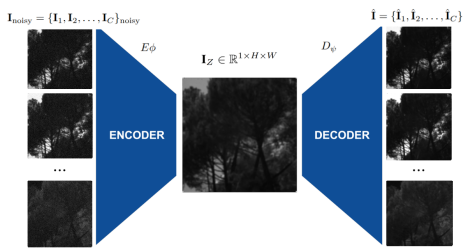


Fig. 3: Proposed multispectral image fusion framework.

Noisy observations are obtained by applying additive Gaussian noise to each channel:

$$\mathbf{I}_{\text{noisy}} = \mathbf{I} + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \tag{1}$$

where σ controls the noise level. The encoder $E_{\phi_{\text{img}}}$, inspired by the ConvNeXt architecture [25], projects the input to an intermediate feature space and applies L generic convolutional blocks with normalization, pointwise channel mixing, and residual connections. A final projection produces a single-channel latent map

$$\mathbf{I}_Z = E_{\phi_{\text{img}}}(\mathbf{I}_{\text{noisy}}) \in \mathbb{R}^{B\times 1\times H\times W}, \tag{2}$$

enforcing a $C \to 1$ compression while preserving spatial resolution. The decoder $D_{\psi_{\text{img}}}$ reconstructs the denoised output $\widehat{\mathbf{I}} \in \mathbb{R}^{B\times C\times H\times W}$ from $\mathbf{I}_Z$ using a sequence of generic transpose and standard convolutions with nonlinear activations:

$$\widehat{\mathbf{I}} = D_{\psi_{\text{img}}}(\mathbf{I}_Z). \tag{3}$$

All convolutional and linear layers are initialized with appropriate weights and zero biases. The specific architectural design, including convolutional block types, channel widths, and other hyperparameters, is provided in Sec. IV-B. This architecture yields a compact latent representation while enabling accurate reconstruction of multispectral inputs.

C. Autoencoder Training and Inference

The proposed model is trained in a denoising autoencoder (DAE) paradigm. Noisy observations $\mathbf{I}_{\text{noisy}}$ are generated by adding noise to clean multispectral images $\mathbf{I}$. During training, the image encoder

$$E_{\phi_{\text{img}}} : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{1\times H\times W}$$

maps $\mathbf{I}_{\text{noisy}}$ to a fused latent representation

$$\mathbf{I}_Z = E_{\phi_{\text{img}}}(\mathbf{I}_{\text{noisy}}),$$

and the image decoder

$$D_{\psi_{\text{img}}} : \mathbb{R}^{1\times H\times W} \to \mathbb{R}^{C\times H\times W}$$

reconstructs a denoised estimate

$$\widehat{\mathbf{I}} = D_{\psi_{\text{img}}}(\mathbf{I}_Z).$$

The encoder-decoder parameters $\theta_{\text{img}} = \{\phi_{\text{img}}, \psi_{\text{img}}\}$ are optimized by minimizing the mean squared error (MSE) loss:

$$\min_{\theta_{\text{img}}} \mathcal{L}_{\text{img}}(\theta_{\text{img}}) = \frac{1}{N}\sum_{i=1}^{N} \left\| D_{\psi_{\text{img}}}\left(E_{\phi_{\text{img}}}(\mathbf{I}^{(i)}_{\text{noisy}})\right) - \mathbf{I}^{(i)} \right\|_2^2. \tag{4}$$

At inference, only the encoder is retained to provide a fused, noise-attenuated representation. Given a noisy input $\mathbf{I}_{\text{noisy}}$, the image encoder outputs the fused intensity map

$$\mathbf{I}_Z = E_{\phi_{\text{img}}}(\mathbf{I}_{\text{noisy}}).$$

This fused map serves as a compact, noise-robust representation suitable for downstream tasks.

D. Auxiliary Gradient Autoencoders

To capture edge structure directly in the fused domain, we introduce two auxiliary autoencoders, one for the horizontal (u) and one for the vertical (v) direction. Unlike computing gradients from the fused intensity map $\mathbf{I}_Z$ after fusion, these branches are trained to produce fused gradient maps as their latent outputs. Let $\nabla\mathbf{I}[u], \nabla\mathbf{I}[v]$ denote the horizontal and vertical discrete gradients of an image, applied channel-wise. From the noisy inputs $\mathbf{I}_{\text{noisy}}$, we construct $\mathbf{I}_{\text{noisy},u} = \nabla\mathbf{I}_{\text{noisy}}[u]$ and $\mathbf{I}_{\text{noisy},v} = \nabla\mathbf{I}_{\text{noisy}}[v]$, with corresponding clean targets $\mathbf{I}_u = \nabla\mathbf{I}[u]$ and $\mathbf{I}_v = \nabla\mathbf{I}[v]$. The horizontal gradient autoencoder is parameterized by (E_{ϕ_u}, D_{ψ_u}), and the vertical one by (E_{ϕ_v}, D_{ψ_v}). They encode noisy gradient inputs into single–channel fused gradient latents

$$\mathbf{I}_{Z,u} = E_{\phi_u}(\mathbf{I}_{\text{noisy},u}), \qquad \mathbf{I}_{Z,v} = E_{\phi_v}(\mathbf{I}_{\text{noisy},v}), \tag{5}$$

and reconstruct denoised gradient estimates as

$$\widehat{\mathbf{I}}_u = D_{\psi_u}(\mathbf{I}_{Z,u}) \approx \nabla\mathbf{I}[u], \qquad \widehat{\mathbf{I}}_v = D_{\psi_v}(\mathbf{I}_{Z,v}) \approx \nabla\mathbf{I}[v]. \tag{6}$$

Each auxiliary model minimizes an MSE objective analogous to the image fusion branch:

$$\mathcal{L}_u = \tfrac{1}{N}\sum_{i=1}^{N} \left\| D_{\psi_u}(E_{\phi_u}(\nabla\mathbf{I}^{(i)}_{\text{noisy}}[u])) - \nabla\mathbf{I}^{(i)}[u] \right\|_2^2, \tag{7}$$

$$\mathcal{L}_v = \tfrac{1}{N}\sum_{i=1}^{N} \left\| D_{\psi_v}(E_{\phi_v}(\nabla\mathbf{I}^{(i)}_{\text{noisy}}[v])) - \nabla\mathbf{I}^{(i)}[v] \right\|_2^2. \tag{8}$$

The auxiliary autoencoders adopt the same architecture, optimization strategy, and parameter initialization as the main fusion autoencoder $(E_{\phi_{\text{img}}}, D_{\psi_{\text{img}}})$, and their training and inference follow the same strategy described in Sec. III-C. It is important to note that the image autoencoder and the two gradient autoencoders are trained independently, each minimizing its own objective $\mathcal{L}_{\text{img}}, \mathcal{L}_u$, or $\mathcal{L}_v$.

IV. Simulation and Experiments Results

A. Simulation Setup

To evaluate the proposed learning-based image fusion strategy within the Multi-Spectral Visual Servoing (MSVS) framework, we conducted a series of simulations using a multispectral imaging system. The control objective in all cases is to drive a six-degree-of-freedom (6-DoF) camera so that the acquired image matches the reference image captured at a desired pose, following the error definition introduced in the background section. The camera considered in this study is the TOUCAN multispectral sensor [26], which provides data across ten distinct spectral bands. This wide spectral coverage makes it particularly suitable for assessing the robustness of MSVS in diverse environments. To obtain a compact representation, two strategies are compared: (i) the Pixel Selection method [5], serving as the baseline, and (ii) the proposed learning-based method using autoencoders. An illustration of this conversion is provided in Fig. 4, where the first ten images correspond to the raw spectral bands, and the final outputs show the resulting fused 2D images for both methods.

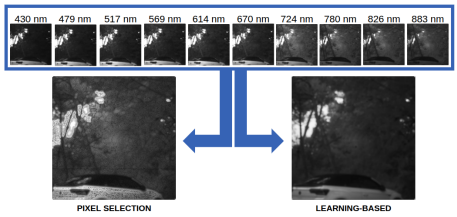


Fig. 4: From multispectral to 2D image.

The simulation environment is built around a virtual camera model. All control and image processing tasks run on a Linux workstation, a 14-core Intel Alder Lake, 3.0 GHz, leveraging the OPENROX open-source library [27]. Each Image has a resolution of 512×512 pixels to guarantee sufficient analytical detail. The multispectral datasets were recorded in diverse outdoor settings to capture environmental difficulties, deliberately testing the system against illumination changing and complex textures. Both the proposed and baseline methods are evaluated in terms of convergence behavior, positioning accuracy, and robustness to Gaussian noise, reflecting the stability and error minimization objectives of MSVS. The quantitative results of these comparisons are reported in the following section.

B. Autoencoder Experimental Details

We use a dataset of 30 coregistered multispectral images, each consisting of 10 spectral channels with a resolution of 512×512 pixels. The data are split into 20 images for training, 5 for validation, and 5 for testing, ensuring no scene overlap. Although the number of training samples is limited, each full-resolution multispectral image provides a large number of spatial–spectral samples. Moreover, the lightweight autoencoder architecture mitigates overfitting and enables stable training from scratch, yielding consistent validation and test performance. Training is performed in a self-supervised (denoising) manner, where noisy observations

$$\mathbf{I}_{\text{noisy}} = \mathbf{I} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}),$$

are generated by applying additive Gaussian noise to the clean images $\mathbf{I}$. The injected noise is not intended to exactly model real-world disturbance statistics. Instead, additive Gaussian noise with an empirically chosen standard deviation of $\sigma = 0.3$ (assuming normalized intensities) is used as a simplified approximation of dominant additive sensor noise sources (e.g., electronic readout and thermal noise), enabling robust denoising autoencoder training while preserving spatial and spectral structure. No human-labeled ground truths are required.

The autoencoder follows an encoder-decoder design inspired by ConvNeXt [25] architecture. The encoder begins with a 3×3 stem convolution (stride 1, padding 1) projecting the input to an intermediate width, followed by $L = 4$ ConvNeXt blocks that use depthwise convolutions, LayerNorm, pointwise 1×1 channel mixing, and residual paths with learnable layer scaling and stochastic depth, producing

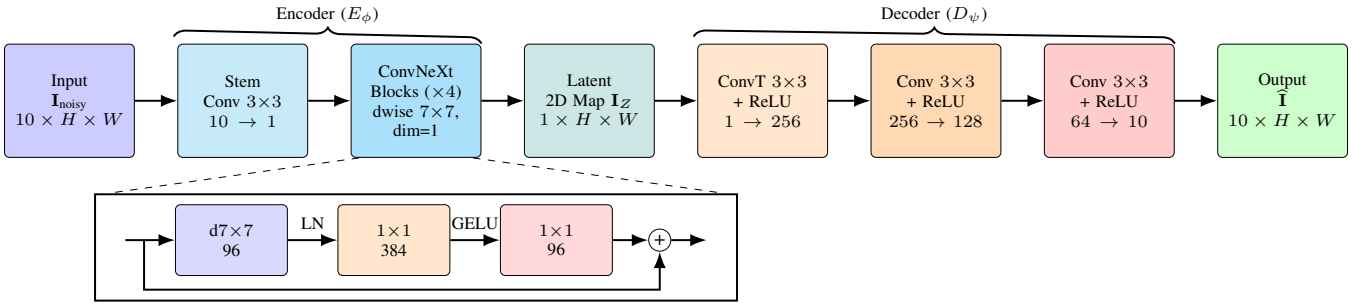


Fig. 5: Autoencoder architecture with expanded ConvNeXt block inset. The encoder E_ϕ applies a 3×3 stem convolution and four ConvNeXt blocks (depthwise 7×7, LayerNorm, Gaussian Error Linear Unit (GELU), pointwise convolutions) to produce a latent map $\mathbf{I}_Z$. The inset shows a single ConvNeXt block. The decoder D_ψ reconstructs the clean 10-channel output using 3×3 transpose and standard convolutions with ReLU activations ($1 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 10$).

a single–channel latent map $\mathbf{I}_Z \in \mathbb{R}^{B\times 1\times H\times W}$, enforcing a $10 \rightarrow 1$ compression at native spatial resolution.

The decoder reconstructs the denoised output $\widehat{\mathbf{I}} \in \mathbb{R}^{B\times 10\times H\times W}$ from $\mathbf{I}_Z$ using a sequence of transpose and standard convolutions with ReLU activations; the channel schedule is $1\rightarrow 256\rightarrow 128\rightarrow 64\rightarrow 10$, restoring the original 10-channel structure. All convolutional and linear layers are initialized with truncated normal weights (std 0.02) and zero biases. The detailed architecture is given in Fig. 5.

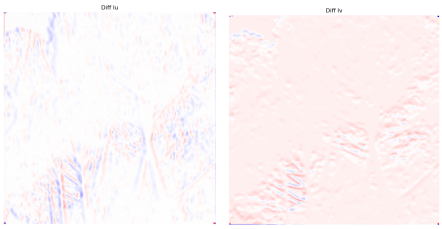
The model is trained for 100 epochs using the AdamW optimizer with a learning rate of 5×10^{-4} and a batch size of 1. The validation set is used for hyperparameter selection. The same training procedure is employed for both the image and gradient autoencoders.

C. Simulation Results

We designed a simulation phase built around the Multi-Spectral Visual Servoing (MSVS) framework. We chose the planar Direct Visual Servoing (DVS) formulation from [28], which assumes the scene is locally flat. Minimizing the photometric error between the current and reference 2D fused images is the key to controlling the problem. For the control side, we used standard proportional gains across all six degrees of freedom, setting $\gamma_v = 0.1$ for translations and $\gamma_\omega = 0.05$ for rotations.

In the virtual environment, every trial starts with a deliberate and consistent misalignment generated by applying a specific transformation matrix to the reference image. To evaluate robustness against large displacements, the camera is subjected to a translation of 0.4 meters along the z-axis combined with a 150° rotation.

The performance of the two MSVS variants is then compared across four distinct scenarios. To emulate real-world disturbances, Gaussian noise with mean 0.05 and variance 0.020 is injected into the multispectral data. The noise is applied independently to each spectral band, thereby simulating environmental variability and sensor imperfections. This perturbation allows for a systematic evaluation of the algorithm's resilience and its ability to maintain stable convergence under degraded conditions. An additional aspect concerns the computational feasibility of the Learning-Based approach. At each iteration, the multispectral cube is fed to the autoencoder, which reconstructs both fused 2D image and its horizontal and vertical gradients. While this provides accurate gradient information, it requires approximately 150 ms per cycle, in addition to the time needed for the visual servoing control computation, which may compromise real-time applicability. To reduce this overhead, the alternative strategy of computing the gradients directly from the generated 2D image is evaluated. The two sets of gradients, the autoencoder-generated and computed ones, were found to be nearly identical. The differences are evaluated using the Root Mean Squared Error (RMSE) metric. In the example in Fig. 6, for the horizontal gradient the RMSE is 0.039 and for the vertical gradient it is 0.064. These values indicate that the two image gradients are reproduced with good fidelity, particularly for the horizontal direction. The errors are mainly localized around high-contrast regions, in particular on spots corresponding to the sky, while most of the image remains well reconstructed.

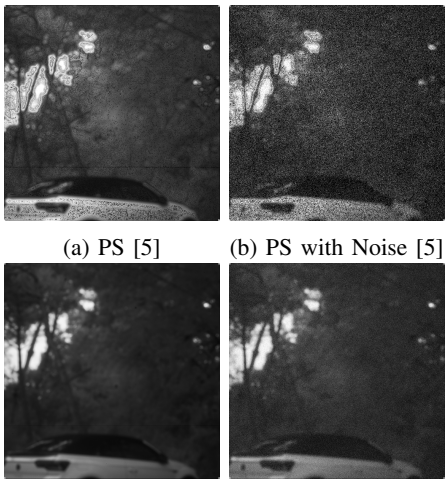


(a) $\nabla \mathbf{I}_u$ difference (b) $\nabla \mathbf{I}_v$ difference

Fig. 6: Difference between the gradients generated by the autoencoder and those computed from the fused image.

Given these results, the gradient directly computed from the fused image is used, which reduces the generation time to about 60–70 ms per cycle while preserving precision. In the first set of positioning simulations, the snapshot was captured in a vegetation-rich environment, with the objective of aligning the image to a partially visible vehicle. In Fig. 7, it is possible to observe the first advantage of the Learning-Based approach over the Pixel Selection method. When no noise is injected, the images constructed by both methods are quite similar, as shown in Fig. 7a and 7c capturing the most relevant features given by the multispectral camera.

However, when noise is introduced, the Pixel Selection method struggles to maintain the integrity of the tree trunk's features, as seen in Fig. 7b. In contrast, the Learning-Based method demonstrates a remarkable ability to preserve these critical features despite the noise thanks the denoise of the autoencoder, as illustrated in Fig. 7d. This robustness is crucial for effective Visual Servoing, as it ensures that the control system can rely on consistent and more robust visual cues even in the presence of significant disturbances.



(a) PS [5] (b) PS with Noise [5]

(c) LB (d) LB with Noise

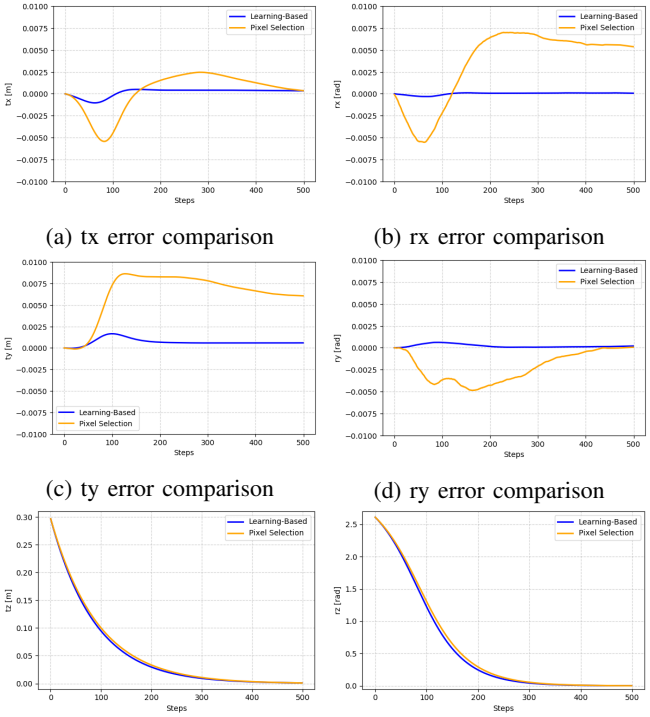
Fig. 7: Comparison of the two methods in the first scenario.

The reference image, initial displacement, and final result after convergence for the Learning-Based method with noise are shown in Fig. 8.

(a) Reference (b) Initial (c) Final

Fig. 8: MSVS simulation using first scenario.

In the absence of noise, the convergence behavior and final positioning accuracy of both methods are virtually indistinguishable. The contrast only becomes clear once noise is introduced. As illustrated in Fig. 9, the Pixel Selection (PS) baseline degrades significantly, its translational error begins to oscillate, extending the convergence up to 500 iterations. Our learning-based approach handles better noisy conditions. In fact, it traces the exact same stable trajectory as it did on the clean data, settling in just 400 iterations. What makes this practically useful for field MSVS is that this added resilience does not carry a computational penalty. While the per-cycle runtimes are roughly equal for both methods, the autoencoder produces inherently smoother images, allowing our system to reach convergence in significantly fewer overall steps.



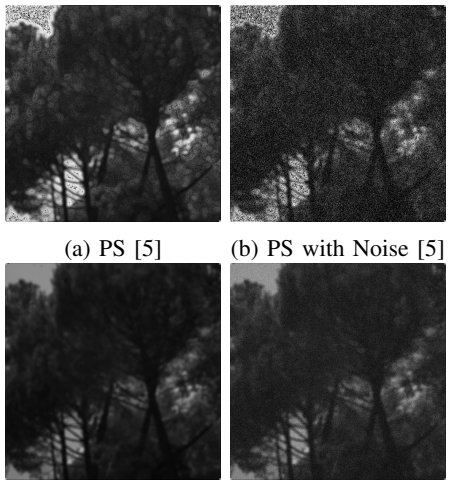
(a) tx error comparison (b) rx error comparison

(c) ty error comparison (d) ry error comparison

(e) tz error comparison (f) rz error comparison

Fig. 9: Convergence plots for the first scenario with noise.

In the second scenario, the snapshot captures a forest environment with a focus on the trees. This scene presents a more complex texture compared the previous scenario, challenging the Visual Servoing algorithms to accurately interpret and respond to the intricate details. Thanks to the multispectral details, both methods can effectively capture the rich features of the vegetation. However, the Learning-Based method again demonstrates superior robustness in the presence of noise, as shown in Fig. 10.



(a) PS [5] (b) PS with Noise [5]

(c) LB (d) LB with Noise

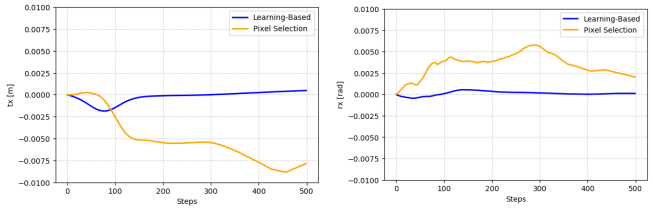
Fig. 10: Comparison PS and LB in vegetation scenario.

The reference image, initial displacement, and final result after convergence for the Learning-Based method with noise are shown in Fig. 11.

(a) Reference (b) Initial (c) Final

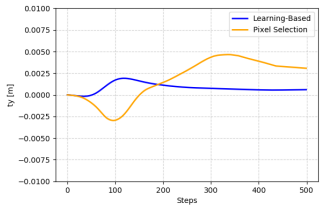
Fig. 11: MSVS simulation using vegetation scenario.

Similar to the first scenario, the convergence and performance of both methods are comparable in the absence of noise. However, the introduction of noise reveals the limitations of the Pixel Selection method, which exhibits increased oscillations and a slower convergence rate, with 500 iterations, as depicted in Fig. 12. In contrast, the Learning-Based approach maintains a stable and efficient convergence trajectory, underscoring its effectiveness in handling complex textures under noisy conditions, taking just 400 iterations.

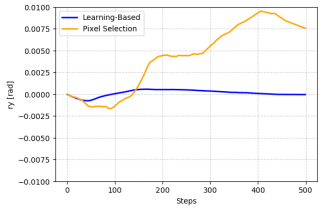


(a) tx error comparison

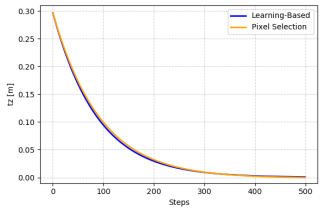
(b) rx error comparison



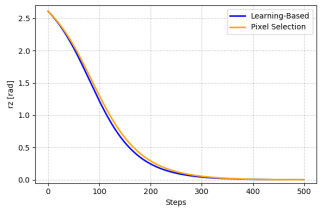
(c) ty error comparison



(d) ry error comparison



(e) tz error comparison



(f) rz error comparison

Fig. 12: Convergence plots for the second scenario with noise.

Similar results are observed in the other scenarios. In each case, the Learning-Based method consistently outperformed the Pixel Selection approach under noisy conditions, achieving stable convergence. Across all scenarios, the Learning-Based approach not only ensured more stable convergence under noisy conditions but also maintained computational feasibility, with runtimes comparable to the Pixel Selection strategy. In particular, as it can be seen in the Table I, while Pixel Selection required about 170 ms per control cycle, including 10 ms to extract a 2D image from the multispectral cube, the Learning-Based method achieved slightly faster cycles of 150–160 ms. Although generating the 2D image through the autoencoder incurs a higher cost, 60–70 ms, the improved image quality leads to quicker convergence and fewer iterations, offsetting the added computation. This combination of robustness and efficiency makes the Learning-Based strategy a strong candidate for real-time MSVS applications.

Method	Img Generation	Total Time per Cycle
Pixel Selection [5]	**10** ms	170 ms
Learning-Based	60-70 ms	**150-160** ms

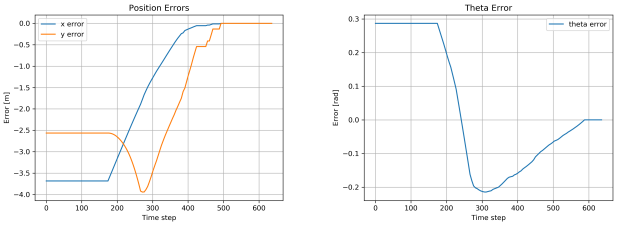
TABLE I: Computational time comparison between Pixel Selection and Learning-Based methods.

D. Real-world Experiments

Outdoor and indoor experiments were conducted near the laboratory to validate the proposed method in real conditions. The robotic platform used is the SCOUT MINI AgileX, operating via ROS. Equipped with a mounted multispectral camera, the robot's task was to seamlessly align its current view with an offline reference image taken at a known location. We managed the entire setup from the same PC used in our simulations to ensure a fair comparison across conditions. During these experiments, we applied an adaptive gain strategy. By starting with the simulated gain value and gradually increasing it as the robot closed the distance to the target, we maintained travel stability during wide displacements while drastically speeding up convergence at the end.

(a) Reference (b) Initial (c) Final

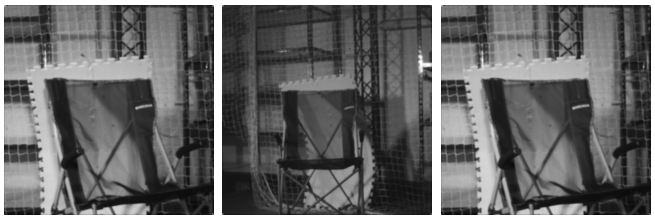
Fig. 13: First real-world scenario experiments.



(a) Position error (b) Rotational error

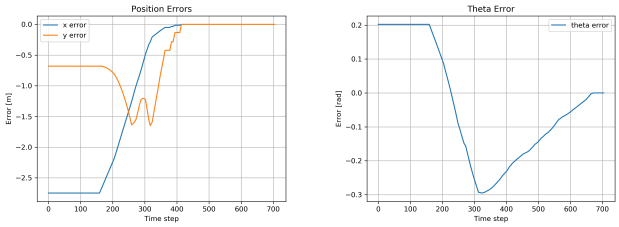
Fig. 14: First real-world scenario MSVS results.

Ultimately, the system achieved accurate alignment in every single trial, regardless of where the robot was initially positioned. Fig.13 and Fig.15 highlight these successful hardware alignments under distinctly different lighting and environmental conditions. Furthermore, the error plots detailed in Fig.14 and Fig.16 track a steady convergence to zero for both position and orientation, proving the framework's practical reliability.



(a) Reference (b) Initial (c) Final

Fig. 15: Second real-world scenario experiments.



(a) Position error (b) Rotational error

Fig. 16: Second real-world scenario MSVS results.

V. Conclusions

This paper presented a Hybrid Learning-Based Multi-Spectral Visual Servoing (HL-MSVS) approach in which an autoencoder is used to fuse multispectral data into a compact 2D representation, which is then used as input to a standard rule-based Direct Visual Servoing scheme. By learning to attenuate noise and produce smoother fused images, the proposed strategy improves the reliability of visual feedback for control. Simulation results show that while both the literature and our proposed method perform similarly in noise-free conditions, the learning-based approach exhibits significantly greater resilience under noisy scenarios, maintaining stable convergence and reducing oscillations in the control error. Importantly, these benefits are achieved without compromising computational efficiency, making the method suitable for real-time Visual Servoing in natural and unstructured environments. Real-world experiments further validate the effectiveness of the learning-based fusion, confirming its practical applicability beyond simulation.

Acknowledgment

This work has been supported by the French government, through the UCA DS4H Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-17-EURE-0004. We would like to thank the engineer Louis Verduci for his code development for efficient multispectral camera data capture in the framework of RobForRisk.

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE RAM*, 2006.

[2] C.-I. Chang, Q. Du, T.-L. Sun, and M. Althouse, "A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification," *IEEE GRS*, 1999.

[3] W. Sun, W. Li, J. Li, and Y. Lai, "Band selection using sparse nonnegative matrix factorization with the thresholded earth's mover distance for hyperspectral imagery classification," *Earth Science Informatics*, 2015.

[4] X. Cao, B. Wu, D. Tao, and L. Jiao, "Automatic band selection using spatial-structure information and classifier-based clustering," *IEEE JSTARS*, 2016.

[5] E. Fiasché, E. Malis, and P. Martinet, "Multi-spectral visual servoing," in *IEEE/RSJ IROS*, 2024.

[6] J. Ma, L. Tang, M. Xu, H. Zhang, and G. Xiao, "STDFusionNet: An infrared and visible image fusion network based on salient target detection," *IEEE Transactions on Instrumentation and Measurement*, 2021.

[7] J. Yang, Y.-Q. Zhao, and J. C.-W. Chan, "Hyperspectral and multispectral image fusion via deep two-branches convolutional neural network," *Remote Sensing*, 2018.

[8] Q. Xie, M. Zhou, Q. Zhao, D. Meng, W. Zuo, and Z. Xu, "Multispectral and hyperspectral image fusion by MS/HS fusion net," in *IEEE/CVF CVPR*, 2019.

[9] J. Zhu, H. Wang, Y. Xu, Z. Wu, and Z. Wei, "Self-learning hyperspectral and multispectral image fusion via adaptive residual guided subspace diffusion model," in *IEEE/CVF CVPR*, 2025.

[10] D. Bank, N. Koenigstein, and R. Giryes, *Autoencoders*. Springer International Publishing, 2023.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[12] A. Azarang, H. E. Manoochehri, and N. Kehtarnavaz, "Convolutional autoencoder-based multispectral image fusion," *IEEE Access*, 2019.

[13] D. Liu, J. Cheng, and J. Ma, "Model-inspired autoencoder for unsupervised hyperspectral image super-resolution," *IEEE TIP*, 2021.

[14] E. Alfaro-Mej'ıa, V. Manian, J. D. Ortiz, and R. P. Tokars, "A blind convolutional deep autoencoder for spectral unmixing of hyperspectral images over waterbodies," *Frontiers in Earth Science*, 2023.

[15] F. Castelli, S. Michieletto, S. Ghidoni, and E. Pagello, "A machine learning-based visual servoing approach for fast robot control in industrial setting," *International Journal of Advanced Robotic Systems*, 2017.

[16] E. G. Ribeiro, R. de Queiroz Mendes, and V. Grassi, "Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation," *Robotics and Autonomous Systems*, 2021.

[17] Z. Li, Y. Zhou, L. Wang, X. Zhang, A. Li, M. Zhu, and Q. Wu, "An end-to-end controller with image-based visual servoing of industrial manipulators with soft-actor-critic algorithm," *Knowledge-Based Systems*, p. 112980, 2025.

[18] J. Gao, Y. He, Y. Chen, and Y. Li, "Learning end-to-end visual servoing using an improved soft actor-critic approach with centralized novelty measurement," *IEEE TIM*, 2023.

[19] A. Saxena, H. Pandya, G. Kumar, A. Gaud, and K. M. Krishna, "Exploring convolutional networks for end-to-end visual servoing," in *IEEE ICRA*, 2017.

[20] A.-P. Botezatu and A. Burlacu, "A short review of deep learning methods in visual servoing systems," *Bulletin of the Polytechnic Institute of Iași. Electrical Engineering, Power Engineering, Electronics Section*, 2023.

[21] Y. He, J. Gao, and Y. Chen, "Deep learning-based pose prediction for visual servoing of robotic manipulators using image similarity," *Neurocomputing*, 2022.

[22] A. Al-Shanoon and H. Lang, "Robotic manipulation based on 3-d visual servoing and deep neural networks," *Robotics and Autonomous Systems*, 2022.

[23] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *IEEE ICRA*, 2018.

[24] K. Ahlin, B. Joffe, A.-P. Hu, G. McMurray, and N. Sadegh, "Autonomous leaf picking using deep learning and visual-servoing," *IFAC-PapersOnLine*, 2016.

[25] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *IEEE/CVF CVPR*, 2022.

[26] S. Tisserand, "Vis-nir hyperspectral cameras," *Photoniques*, 2021.

[27] Openrox open-source c library providing real-time algorithms for robotics. [Online]. Available: https://github.com/ACENTAURI-INRIA/openrox

[28] E. Malis and S. Benhimane, "A unified approach to visual tracking and servoing," *Robotics and Autonomous Systems*, 2005.