

DeepMPCVS: Deep Model Predictive Control for Visual Servoing

Pushkal Katara^{*1}, Y V S Harish¹, Harit Pandya², Abhinav Gupta¹, Aadil Mehdi Sanchawala¹, Gourav Kumar³, Brojeshwar Bhowmick³, and K. Madhava Krishna¹

¹Robotics Research Center, IIT Hyderabad, India

²University Of Lincoln, UK

³TCS Innovation Labs, Kolkata, India

Abstract: The simplicity of the visual servoing approach makes it an attractive option for tasks dealing with vision-based control of robots in many real-world applications. However, attaining precise alignment for unseen environments pose a challenge to existing visual servoing approaches. While classical approaches assume a perfect world, the recent data-driven approaches face issues when generalizing to novel environments. In this paper, we aim to combine the best of both worlds. We present a deep model predictive visual servoing framework that can achieve precise alignment with optimal trajectories and can generalize to novel environments. Our framework consists of a deep network for optical flow predictions, which are used along with a predictive model to forecast future optical flow. For generating an optimal set of velocities we present a control network that can be trained on-the-fly without any supervision. Through extensive simulations on photo-realistic indoor settings of the popular Habitat framework, we show significant performance gain due to the proposed formulation vis-a-vis recent state of the art methods. Specifically, we show a faster convergence and an improved performance in trajectory length over recent approaches.

Keywords: Visual Servoing, Deep MPC

1 Introduction

The concept of visual servoing has been widely used in the field of robotics for a variety of tasks from manipulation to navigation [1, 2, 3, 4, 5]. Mapping the observation space to the control/action space is the fundamental objective behind most of the visual servoing problems. Classically, visual servoing is highly dependent on handcrafted image features, environment modeling, and an accurate understanding of system dynamics for the generation of control commands [6]. Sensitivity to the accuracy of the available information also prevents such systems from adjusting to the statistical irregularities of the world. To mitigate these issues, recent approaches aim to learn supervised models that predict relative camera poses between the current and the desired images [7, 8, 9]. Although this overcomes the requirements of having handcrafted image features or any explicit information about the environment, it fails to generalize in unseen environments. Other approaches further attempt to exploit information from intermediate representations like optical flow [10], however, they use classical visual servoing controller which only looks one-step ahead hence results in sub-optimal trajectories. On the other hand, many deep reinforcement learning based approaches such as [11] and [12] do not show their convergence in a 6-DoF domain due to an intractable number of samples in higher dimensional actions and continuous state space. This narrows down their scope of performance in real-world visual servoing tasks for precise alignment. Furthermore, they also need to be extensively retrained in newer environments as a result, their generalisation capabilities are

*: katarapushkal@gmail.com

limited. A few approaches [13, 14], employ a model based learning framework for visual servoing, they attempt to learn a forward predictive model to represent the process and a policy for planning over the predictive model. However, their predictive model is trained to minimise the appearance loss which is difficult to optimise for a longer horizon and does not generalize to novel environments. Furthermore, the policy is obtained either through sampling [13] or by learning from human demonstrations [14] that do not scale to a higher dimensional action space.

In this work, we present a deep model predictive control strategy that exploits the visual servoing concept in a principled fashion. This is achieved by formulating a novel state prediction model for visual servoing based on dense off-the-shelf/unsupervised optical flow predictions. The prediction model is employed to generate optimal velocity commands using a control network, we further propose an efficient unsupervised strategy for online optimisation of control network. We validate our approach on a photo-realistic visual servoing dataset proposed in [10], comparing against an exhaustive set of baselines among classical, deep supervised and model predictive visual servoing approaches.

Our contributions are summarized as follows:

- We propose a novel unsupervised Deep Model Predictive Control pipeline for visual servoing using optical flow as intermediate representation in 6-DoF.
- The control actions are learned unsupervised as the LSTM decoder strives to predict future controls over a time horizon such that the optical flow accrued as a consequence of these predictions matches the desired flow. The desired flow between the current and desired images is computed either through a supervised [15] or unsupervised [16] framework.
- The online optimization over the MPC framework enables the proposed approach to adapt to any environment even with inaccurate flow predictions and system dynamics information.
- We compare with existing deep visual servoing methods by evaluating on parameters like translation error, rotation error, trajectory length, total no. of iterations and time per iteration. Our approach exhibits significant convergence and is faster than the state of the art deep visual servoing methods [10], [17], [18],[19]. We present both qualitative and quantitative results for the same. We make the code ² publicly available.

2 Related Work

Visual Servoing. Visual Servo (VS) relates to the problem of attaining a goal pose in the environment using image measurements from a camera sensor attached to a robot. It can be broadly classified in two approaches: (i) PBVS: Pose based Visual Servoing where hand-crafted features from images are used to estimate geometrical information such as pose of the target, camera parameters and controller is responsible to minimize pose error between current image and goal image. (ii) IBVS: Image-based Visual Servoing approaches perform feature extraction and minimize the feature error explicitly in image space. The control law is to perform gradient descent in feature space, which is then mapped to the robot’s velocities using an Image Jacobian [6]. Classical visual servoing approaches [1, 4, 5, 20] employ local appearance based features such as keypoints, lines and contours to describe the scene. On the other hand, direct visual servoing approaches [19, 21] skip the feature extraction and directly minimize the difference in images. Although these direct approaches mitigate the issue of incorrect matches, they do not converge for larger camera transformations. One of the limitations of classical visual servoing methods is the requirement of the depth of the visual features while computing the Image Jacobian, which is generally not available in the case of a monocular camera.

Deep Visual Servoing. Recent deep learning driven approaches [7, 8, 18] learn deep neural networks that directly aim to predict the relative camera pose between the current and the desired image. The controller then takes a small step to minimize this error. These neural networks are learned in a supervised fashion and consequently do not generalize well to unseen environments. A very recent approach [10] combines deep learning with visual servoing by learning intermediate flow representations and minimizing their difference using a classical visual servoing controller.

²Github Link: <https://github.com/pushkalkatara/DeepMPCVS>

This approach effectively generalizes to novel environments in 6-DoF. However, since the controller optimizes only for a single time step, it generates sub-optimal trajectories and could get stuck in local minima as a result of this greedy behavior.

Visual Navigation using Reinforcement Learning. Visual servoing could also be posed as a target driven visual navigation problem. Motivated from the success attained by deep reinforcement learning (DRL) approaches in maze navigation tasks, there has been an increase in interest in applying imitation learning [22] or reinforcement learning methods for visual navigation in indoor environments [11, 12, 23, 24, 25]. In contrast to classical visual servoing approaches that greedily plan one time step, DRL approaches plan an optimal policy over a longer horizon even when there is no overlap between the current and desired image. However, such model free end-to-end learning based approaches are sample inefficient, thus do not scale to higher dimensional continuous actions and face difficulty while generalising to novel environments.

Model Based Visual Control. Model predictive control (MPC) approaches have been successful in learning complex skills in robotics such as controlling quadrotors [26], and humanoids [27] using accurate system dynamics. Keypoints based MPC models proposed by [28, 29] have been previously used in IBVS to generate optimal policies under the assumption of accurate matching and using a handful of keypoints. Classical MPC approaches have two limitations: firstly, they require accurate dynamics, secondly, they do not scale well in large state spaces. Recent deep learning approaches aim to improve upon this limitation by simultaneously learning the model dynamics along with the policies [30, 31]. For learning visio-motor control, recent approaches propose a deep network to learn forward dynamics by predicting future observations [13, 14]. To learn the policies along with dynamics, there are a few choices: (i) projecting the latent space into a Cartesian world and classical approaches to generate control [32, 33]. However, directly learning camera pose could be inaccurate for unseen environments and 6-DoF actions. (ii) Hirose et al. [14] use a neural network to approximate policy which is learned in a supervised fashion from human demonstrations. However, the issue with this method is obtaining a large and diverse dataset of human demonstrations which is expensive. Another approach [13] uses visual MPC where model is trained using collected data, further perform MPC optimisation using CEM which combined makes the approach supervised and sampling-dependent making it less generalizable and less efficient than the proposed method.

This work bridges the gap between classical and learning-based control for 6-DoF image-based visual servoing (IBVS) in novel environments with continuous action and state space. We select optical flow to represent our states instead of directly working with images. The dense optical flow is predicted using deep neural networks. Subsequently, we reformulate a predictive model for forecasting the evolution of states given a sequence of actions (velocities). We then learn a recurrent control network on-the-fly in an unsupervised fashion for computing an optimal set of velocity for a given goal state. We show superior performance vis-a-vis Deep Visual Servoing methods due to a receding horizon controller even as the framework generalizes to new environments without needing to retrain or finetune. The controller regresses to a continuous space of outputs over 6-DoF.

3 Approach

In this work, we consider the problem of visual servoing as a target driven navigation in unseen environments using a monocular camera. Given I_t , the observation of the robot in the form of monocular RGB image at any time instant t and the desired observation I^* , our goal is to generate optimal control commands $[v_t, \omega_t]$ in 6-DoF that minimizes the photometric error between I^* and I_t . We assume that the camera is attached to the robot (also known as eye-in-hand configuration) and is calibrated. We further assume that the environment is collision free and there exists a partial overlap between the initial and desired image. In contrast to existing visual navigation approaches [14], we aim to generate trajectories in continuous actions (in 6-DoF) and state space. Figure 1 provides an overview of our approach.

3.1 Learning Intermediate Representation

Instead of directly planning in image observation space, we employ optical flow as an intermediate visual representation for encoding differences in images. Optical flow encodes displacement for every pixel which is more relevant information for feature tracking and matching as compared to pixel intensities. As a result, optical flow has been successfully used in motion estimation [34]

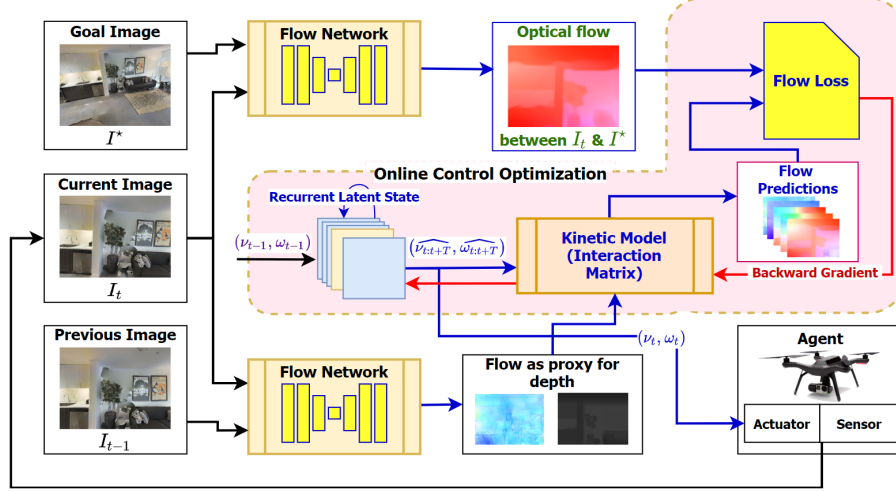


Figure 1: Given agent observation I_t and desired observation I^* the pipeline generates optimal control commands $[v_t, \omega_t]$ for each time-step t until convergence is achieved. The flow network encodes optical flow between I_t and I^* and the control-optimizer architecture learns to generate optimal control for each time-step t in real-time through encoded flow.

and visual servoing [35, 10]. Furthermore, in the presence of translation camera motion in a static environment, dense flow features could also be used for estimating image depth [36]. Although we do not constraint ourselves to any specific network for flow or depth estimation, in this work we use a pre-trained neural network, Flownet 2 [15] for flow estimation without any fine-tuning. To showcase that our approach can achieve similar performance even when trained without any supervision, we replace Flownet 2 with another network architecture DDFlow [16], which is trained from scratch using only images in an unsupervised manner. For specifics regarding training of DDFlow, refer to the supplementary material.

3.2 Predictive model for visual servoing

The fundamental objective of IBVS is to minimize the error in visual features between current and desired images, $e_t = \|s(I_t) - s(I^*)\|$, which is solved by performing gradient descent in feature space, resulting in following control law:

$$\mathbf{v}_t = -\lambda L^+(s(I_t) - s(I^*)) \quad (1)$$

where, $\mathbf{v}_t = [v_t, \omega_t]$ is the velocity command generated by the controller in 6-DoF, $s(I_t)$ and $s(I^*)$ are the visual features extracted from images I_t and I^* respectively. The gradient descent step size is depicted by λ and $(\cdot)^+$ denotes pseudo-inverse operation. The image Jacobian or the interaction matrix mapping the camera velocity to rate of change of features is given by,

$$L(Z_t) = \begin{bmatrix} -1/Z_t & 0 & x/Z_t & xy & -(1+x^2) & y \\ 0 & -1/Z_t & y/Z_t & 1+y^2 & -xy & -x \end{bmatrix}. \quad (2)$$

Where, x and y are the normalized image coordinate and Z_t is the scene's depth at (x, y, t) . For a small time interval δt , IBVS control law, i.e. equation (1) can be reformulated as $s_{t+\delta t} = s_t + L(z)\mathbf{v}_t$, which can be rewritten using optical flow as visual features as,

$$\mathcal{F}(I_t, I_{t+\delta t}) = L(Z_t)\mathbf{v}_t. \quad (3)$$

This equation gives us the predictive model that could be employed to reconstruct the future states (flows in our case) from the current velocity. This also points out that generating an optimal set of actions in terms of velocities $\mathbf{v}_{t+1:t+T}^*$ is a critical step for the convergence of the system towards the desired image. We achieve this task by online trajectory optimization using the predictive model and a control network described in equation (3) as described in section 3.3.

3.3 Online Trajectory Optimization using Neural MPC

Model predictive controllers are proven to work well for a wide variety of tasks in which accurate analytical models are difficult to obtain [30]. A model predictive controller aims to generate a set of optimal control actions $\mathbf{v}_{t+1:t+T}^*$ which minimize a given cost function $C(\hat{X}_{t+1:t+T}, \mathbf{v}_{t+1:t+T})$ over predicted state \hat{X} and control inputs \mathbf{v} for some finite time horizon T . In order to formulate visual servoing as MPC with intermediate flow representations, we define our cost function as mean squared error in flow between any two given images. Then MPC objective,

$$\mathbf{v}_{t+1:t+T}^* = \arg \min_{\mathbf{v}_{t+1:t+T}} \|\mathcal{F}(I_t, I^*) - \widehat{\mathcal{F}}(\mathbf{v}_{t+1:t+T})\| \quad (4)$$

is then to generate a set of velocities $\mathbf{v}_{t+1:t+T}^*$ that minimise the error between the desired flow $\mathcal{F}(I_t, I_t^*)$ predicted by the flow network and the generated flow $\widehat{\mathcal{F}}(\mathbf{v}_{t+1:t+T})$. Exploiting the additive nature of optical flows (i.e. $\mathcal{F}(I_1, I_3) = \mathcal{F}(I_1, I_2) + \mathcal{F}(I_2, I_3)$), the generated flow could be written as:

$$\widehat{\mathcal{F}}(\mathbf{v}_{t+1:t+T}) = \sum_{k=1}^T [L(Z_t)\mathbf{v}_{t+k}] \quad (5)$$

using the predictive model from equation (3). This formulation provides optimal trajectory over a horizon T as compared to classical IBVS controller, equation (1) that greedily optimises a single step. Another advantage of using the proposed approach is significant decrease in computation time as compared to classical IBVS controller, as we avoid the overhead of matrix inversion required for computation of pseudo inverse $L^+ = (L^T L + \mu \text{diag}(L^T L))^{-1}$.

The above formulation allows us to control in continuous 6-DoF action space. However, since the dimensionality of our states (flow representations) are large, classical MPC solvers could not be used to optimise equation (4). To tackle this issue, Finn and Levine [13] use a sampling based Cross Entropy Method (CEM), but the approach is model-based sampling which requires large number of iterations to train and is less generalizable as supervised on a dataset. Another recent approach [14] employs a neural network for policy generation. They train their network using human demonstrations. However, obtaining human demonstrations are expensive especially for 6 DoF. In experiments section 4.1, we explicitly compare with these approaches.

In this work, we use a recurrent neural network (RNN) to generate velocity commands. This choice is more efficient over sampling based approaches such as CEM since, RNN can be directly trained using Back Propagation Through Time (BPTT). Furthermore, for sequence prediction tasks RNN is a natural choice over a feed forward neural network. To train our control network, we employ the predictions from the flow network as targets. Thus, our target remains fixed for online trajectory optimization. Note that our linearized predictive model depends only on the depth of the scene, and assuming depth consistency for a smaller horizon, we can write the MPC objective only in terms of control. Hence, our control network could easily be trained in supervised manner by minimising the flow loss:

$$\mathcal{L}_{flow} = \|\widehat{\mathcal{F}}(\widehat{\mathbf{v}}_{t+1:t+T}) - \mathcal{F}(I_t, I^*)\| = \left\| \sum_{k=1}^T [L(Z_t)\widehat{\mathbf{v}}_{t+k}] - \mathcal{F}(I_t, I^*) \right\| \quad (6)$$

Where, the RNN $g(\mathbf{v}_t, \theta)$ could be used to generate velocity at next time instance $\widehat{\mathbf{v}}_{t+1} = g(\mathbf{v}_t, \theta)$ given the current velocity. Hence, given previous velocity \mathbf{v}_{t-1} and tries to improve the predicted velocity and compensate for approximations in predictive model by adjusting its parameters θ according such that the flow loss is minimized.

3.4 Control Network

In this work, we use a Long-Short-Term Memory architecture (LSTM) [37] with 5 LSTM cell units as RNN for generating velocities. The sequence length (length of planning horizon as T) is a tuning parameter that trades-off between precision and computation time. In this work we select the sequence length $T = 5$. The network receives previous 6-DoF velocity \mathbf{v}_{t-1} as input and predicts a sequence of T velocities $\widehat{\mathbf{v}}_{t:t+T}$ in 6-DoF. This is multiplied by the interaction matrix $L(z)$ to obtain generated flows $\widehat{\mathcal{F}}_{t:t+T}$. The network is trained online in supervised fashion for $M = 100$ iterations by minimizing the flow loss, equation (6), where the target flow is predicted using our flow network.

Algorithm 1 Model Predictive Visual Servoing using Online Optimisation

Require: I^* , ε ▷ Goal Image, convergence threshold
1: Initialize v_0 with random velocity
2: **while** $\|I - I^*\| \leq \varepsilon$ **do** ▷ Convergence criterion
3: $I_t := \text{get-current-obs}()$ ▷ Obtain the current RGB observation from sensor
4: Predict-Flow ($\mathcal{F}(I_t, I^*)$) ▷ Predict target flow using flow network
5: $L_t := \text{compute-interaction-matrix}(\mathcal{F}(I_t, I_{t-1}))$
6: **for** $m = 0 \rightarrow M$ **do** ▷ Online training the control network
7: $\hat{v}_{t:t+T} := \mathbf{g}(v_{t-1}, \theta_m)$ ▷ Velocity predictions from control network
8: $\hat{\mathcal{F}}(\hat{v}_{t+1:t+T}) = \sum_{k=1}^T [L_t(Z_t) \hat{v}_{t+k}]$ ▷ Generate flow using prediction model
9: $\mathcal{L}_{flow} := \|\hat{\mathcal{F}}(\hat{v}_{t+1:t+T}) - \mathcal{F}(I_t, I^*)\|$. ▷ Compute flow Loss
10: $\theta_{m+1} := \theta_m - \eta \nabla \mathcal{L}_{flow}$ ▷ Update control network parameters
11: **end for**
12: $v_{t+1} = \hat{v}_{t+1}$ ▷ Execute the next control command
13: **end while**

After the network is trained for M iterations we execute the v_{t+1} and observe next image I_{t+1} , which is used to predict next target flow $\mathcal{F}(I_{t+1}, I^*)$. Subsequently, the interaction matrix is recomputed $L(Z_{t+1})$ based on updated depth Z_{t+1} using $\mathcal{F}(I_t, I_{t+1})$ as the proxy. The entire approach can be summarised by algorithm 1.

4 Experiments

In our present work, we have explained an online training process which enables the robot to learn optimal control commands on the go thus making our controller performance independent of the environment it is deployed in. To validate this, we show both qualitative and quantitative results on photo realistic 3D baseline environments as proposed in [10]. The benchmark comprises of 10 indoor photo-realistic environments from the Gibson dataset [38] in the Habitat simulation engine [39]. We use a free-flying RGB camera as our agent so that agent can navigate in all 6-DoFs without any constraint. To compare our approach we consider following baselines: **Photometric visual servoing**[19]: PhotoVS is a learning-free classical visual servoing approach that considers raw image intensities as visual features. **Servonet**[18], is a supervised deep learning approach that attempts to predict relative camera pose from a given image pair. We benchmark this approach without any retraining/fine-tuning. **DFVS**[10]: similar to our approach DFVS employs deep flow representations, however they use classical IBVS controller. To evaluate our controller's performance in isolation from flow network and predictive model we also compare against baselines **(a) Fully-connected Neural-Net:** and **(b) Stochastic Optimization [17]:** known as the cross-entropy method. We provide more details on the baseline in supplementary section.

4.1 Simulation results on Benchmark

We report both qualitative (refer figure 2) and quantitative (refer table 1) results on the benchmark. It can be seen from the table 1, our controller outperforms the current state-of-the-art Image Based Visual Servoing technique [10] in total number of iterations while achieving marginally superior pose error. PhotoVS [19] and Servonet [18] were unable to converge in 5 and 6 out of 10 scenes in the benchmark respectively. It can also be seen from comparison with the baseline (a) Fully-connected Neural-network that it gets stuck in local minima and results in larger pose error. Furthermore, CEM results in shorter trajectories but takes significantly large amount of time to converge as compared to our approach which is intuitive since sampling approaches are inefficient but statistically complete. We train our control network for 100 iterations in each MPC optimization step and use Adam Optimiser with learning rate of 0.0001. The number of training iterations can be tuned using Early Stopping methods to achieve the best tradeoff between speed and accuracy for realtime control. We use a "Nvidia Geforce GTX 1080-Ti Pascal" GPU to benchmark these approaches.

4.2 Controller Performance and Trajectory Comparison

We next compare the trajectories taken by other visual servoing methods like [10, 19] and controller performance with a standard feed forward neural network (NN) and CEM [13] to reach the desired

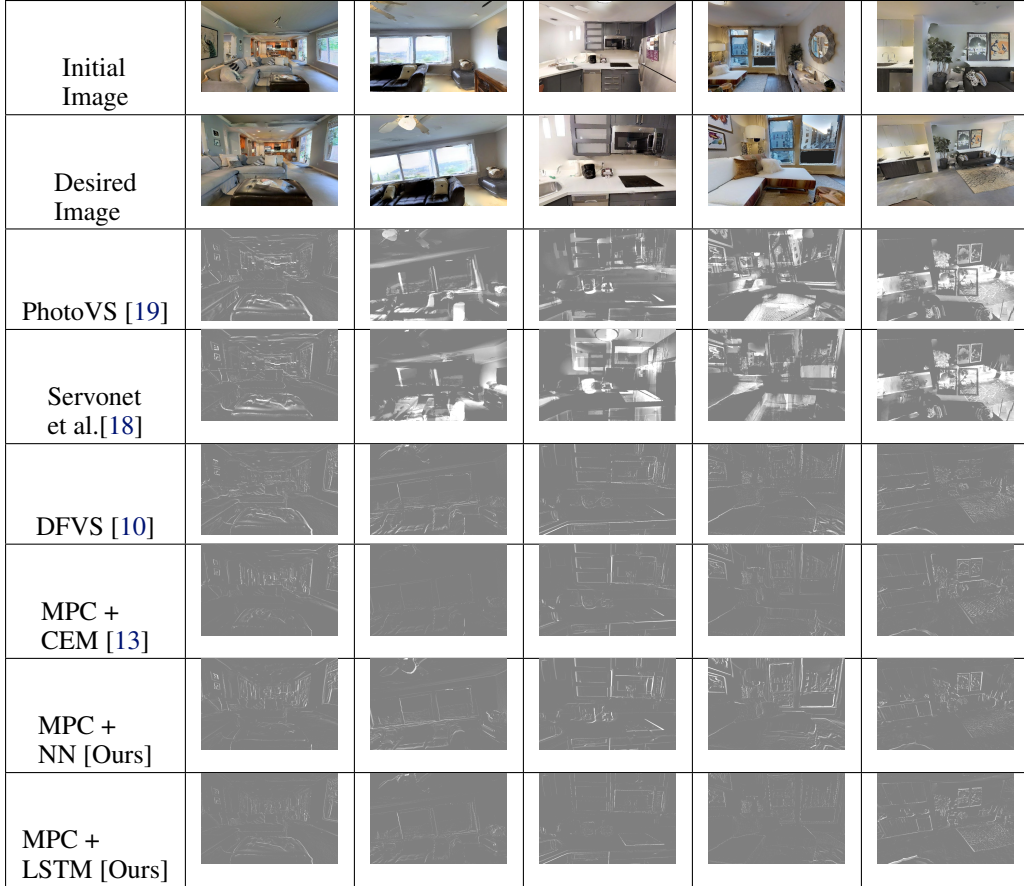


Figure 2: **Qualitative Comparison:** We show the photometric error image representation computed between the desired image I^* and the final image on termination for various approaches. It is to be noted that the termination criteria for the run is photometric error of 500 or less. It can be seen that Saxena et al. [18] and PhotoVS [19] don't converge even with large number of iterations hence showing large photometric errors. We compare Flow-depth[ours] with single-view depth of [10]. Comparisons with MPC + CEM [17] and MPC + NN [Ours] is also shown as their result justify our choice of LSTM based network. We show other qualitative results of left over baselines in supplementary.

Approaches	T. Error (meters)	R. Error (degrees)	Tj. Length (meters)	Iterations
Initial Pose Error	1.6566	21.4166	-	-
DFVS [10]	0.0322	1.7088	1.7277	1009.2222
NN + MPC [ours]	0.102	2.62	1.16*	344*
CEM [17] + MPC	0.0513	0.8644	0.89	859.6667
LSTM + MPC [ours]	0.0296	0.5977	1.081	557.4444

Table 1: **Quantitative Comparison:** We quantitatively compare the average performance for different approaches across the benchmark scenes proposed in [10] and report following metrics: Initial Pose Error, translation error(T. Error), rotation error(R. Error) in degrees and trajectory length(Tj. Length) in meters. This table does not include [19],[18] as they diverge in majority of the environments[5 and 6 respectively of the 10 baseline envs].

goal. The photometric convergence and trajectory plots are shown in fig. 3. It can be observed from the figure that photometric error steadily reduces to less than 500. As seen from the trajectory plots, CEM with predictive model tends to reach the goal in an optimal path followed by our approach, while our approach is taking less MPC steps for convergence.

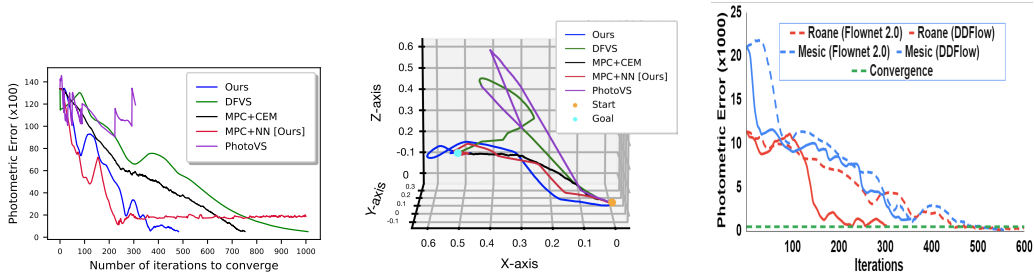


Figure 3: **Controller Performance:** **Left:** Photometric Error (vs) Number of Iterations - Our method [MPC + LSTM] converges in least number of iterations compared to other approaches. **Middle:** Camera Trajectory. **Right:** Controller performance on Unsupervised flow - Dashed lines represents Flownet2 [15] and bold lines represents DDFlow [16]

4.3 Controller performance with Unsupervised flow

To showcase that our deep MPC architecture is also capable of generating optimal control commands even when trained in fully unsupervised manner i.e. here even the flow network is trained without supervision using only images. We use DDFlow [16] instead of Flownet2 [15] to generate the optical flow. For this experiment, we train our flow network offline on two scenes from the benchmark. For the training details we refer the readers to the supplementary material. The photometric error plots in Fig. 3 (Right) shows that our controller successfully converges on both the environments, thereby making our pipeline unsupervised. No ground truth or annotations were provided and the flow was learned in an unsupervised fashion without any unlabeled data.

4.4 Generalisation to Real-World Noise

In order to demonstrate robustness of our approach we perform tests with simulated noise in the Habitat environment. We add Gaussian noise with mean=0.0 and standard deviation=0.1m in all 6-DoF to control commands. We perform test on environment Mesic from our simulation benchmark [10] and show that our pipeline converges to a photometric error of < 500 in 683 iterations in presence of such a large noise, as compared to 504 iterations without noise. This showcases ability to handle actuation noise and generalize in a real world environment.

5 Conclusion and Future Work

In this work, we proposed a deep unsupervised model predictive control architecture for visual servoing in 6-DoF. We demonstrated that our approach can generate control commands in continuous state space which can adapt and perform well despite an inaccurate understanding of the underlying system dynamics. Its ability to be trained in an online fashion makes it easy to adapt well in a completely unknown environment.

Although the proposed approach addresses some of the most crucial challenges and performs better than state-of-the-art methods proposed on this problem, it suffers due to a few drawbacks. Firstly, this approach assumes that the action space of the robot is unobstructed and there is no chance of collision. Secondly, like all of the visual servoing problems, it requires a small minimum overlap between the current and the desired images which constrains this method to act as a local planner. The above limitations provide an opportunity to extend this approach by incorporating methods to handle collision and integrate this method with a long term global planner.

Acknowledgments

This work was done as a collaborative project between International Institute of Information Technology(IIT), Hyderabad India and Tata Consultancy Services(TCS), Innovation Labs, India. This work was supported by research grant from TCS Innovation Labs India and EPSRC UK (project NCNR, National Centre for Nuclear Robotics, EP/R02572X/1)

References

- [1] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: Surfing on the epipoles. *International Journal of Computer Vision*, 1999.
- [2] K. Usher, P. Ridley, and P. Corke. Visual servoing of a car-like vehicle—an application of omnidirectional vision. In *IEEE ICRA*, 2003.
- [3] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck. Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Transactions on Robotics*, 2009.
- [4] M. Jagersand, O. Fuentes, and R. Nelson. Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *ICRA*. IEEE, 1997.
- [5] P. I. Corke. Visual control of robot manipulators—a review. In *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific, 1993.
- [6] F. Chaumette and S. Hutchinson. Visual servo control. i. basic approaches. *IEEE RAM*, 2006.
- [7] C. Yu, Z. Cai, H. Pham, and Q.-C. Pham. Siamese convolutional neural network for sub-millimeter-accurate camera pose estimation and visual servoing. *arXiv preprint arXiv:1903.04713*, 2019.
- [8] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke. Training deep neural networks for visual servoing. In *IEEE ICRA*, 2018.
- [9] C. Sampedro, A. Rodriguez-Ramos, I. Gil, L. Mejias, and P. Campoy. Image-based visual servoing controller for multirotor aerial robots using deep reinforcement learning. In *IEEE/RSJ IROS*, 2018.
- [10] Y. V. S. Harish, H. Pandya, A. Gaud, S. Terupally, S. Shankar, and K. M. Krishna. Dfvs: Deep flow guided scene agnostic image based visual servoing. In *IEEE ICRA*, 2020.
- [11] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning, 2016.
- [12] D. Gordon, A. Kadian, D. Parikh, J. Hoffman, and D. Batra. Splitnet: Sim2sim and task2task transfer for embodied visual navigation, ICCV 2019.
- [13] C. Finn and S. Levine. Deep visual foresight for planning robot motion, ICRA 2016.
- [14] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese. Deep visual mpc-policy learning for navigation. *IEEE Robotics and Automation Letters*, 2019.
- [15] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [16] P. Liu, I. King, M. R. Lyu, and J. Xu. DdfLOW: Learning optical flow with unlabeled data distillation, AAAI, 2019.
- [17] R. Y. Rubinstein and D. P. Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2013.
- [18] 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.
- [19] C. Collewet and E. Marchand. Photometric visual servoing. *IEEE TRO*, 2011.
- [20] E. Malis, G. Chesi, and R. Cipolla. 212d visual servoing with respect to planar contours having complex and unknown shapes. *The International Journal of Robotics Research*, 2003.
- [21] Q. Bateux and E. Marchand. Histograms-based visual servoing. *IEEE Robotics and Automation Letters*, 2017.

- [22] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell. Zero-shot visual imitation, IEEE CVPR workshop, 2018.
- [23] W. Gao, D. Hsu, W. S. Lee, S. Shen, and K. Subramanian. Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation. *arXiv preprint arXiv:1710.05627*, 2017.
- [24] G. J. Stein, C. Bradley, and N. Roy. Learning over subgoals for efficient navigation of structured, unknown environments. In *CoRL*, 2018.
- [25] A. Sax, J. O. Zhang, B. Emi, A. Zamir, S. Savarese, L. Guibas, and J. Malik. Learning to navigate using mid-level visual priors, arXiv preprint arXiv:1912.11121, 2019.
- [26] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In *IEEE ICRA*, 2016.
- [27] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou. Information theoretic mpc for model-based reinforcement learning. In *IEEE ICRA*, 2017.
- [28] M. Sauvee, P. Poignet, E. Dombre, and E. Courtial. Image based visual servoing through nonlinear model predictive control. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1776–1781, 2006.
- [29] G. Allibert, E. Courtial, and F. Chaumette. Predictive control for constrained image-based visual servoing. *IEEE Transactions on Robotics*, 2010.
- [30] I. Lenz, R. A. Knepper, and A. Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, 2015.
- [31] K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz. Deep model predictive control with online learning for complex physical systems, arXiv:1905.10094, 2019.
- [32] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin. Combining optimal control and learning for visual navigation in novel environments. In *Conference on Robot Learning*, 2020.
- [33] P. Drews, G. Williams, B. Goldfain, E. A. Theodorou, and J. M. Rehg. Aggressive deep driving: Combining convolutional neural networks and model predictive control. In *Conference on Robot Learning*, 2017.
- [34] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2010.
- [35] M. Argus, L. Hermann, J. Long, and T. Brox. Flowcontrol: Optical flow based visual servoing, arXiv:2007.00291, 2020.
- [36] B. Shahraray and M. K. Brown. Robust depth estimation from optical flow. In *[1988 Proceedings] Second International Conference on Computer Vision*, 1988.
- [37] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.
- [38] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018.
- [39] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.