Visual Servoing in Autoencoder Latent Space: Supplementary material

Samuel Felton, Pascal Brault, Elisa Fromont, Eric Marchand

Visual Servoing in Autoencoder Latent Space: Supplementary material

Samuel Felton, Pascal Brault, Elisa Fromont, Eric Marchand.

***Abstract*—This document provides additional information about [1], describing further experiments with Autoencoder-based visual servoing (AEVS), a method for visual servoing in the latent space of an autoencoder. Multiple experiments are described, both in a simulated environment and on a real robot. In this document, we first evaluate the impact of the network specification and dataset sizes on the visual servoing results. Then we conduct experiments to ascertain the applicability and transferability of the method to complex scenes.**

I. Result analysis

In order to ascertain the behaviour of our method, we provide in this section more detail on the results obtained on the two test cases, described in section IV-B of the paper [1]. For the look-at test scenario, where the samples are generated so that they look roughly at the same part of the scene, it is interesting to examine the relationship between convergence and image overlap. In this scenario, we first generate the desired pose $\mathbf{r}^*$, with the corresponding camera having a 3D point $\mathbf{X}^*$ of the scene at the center of its view (its focal point). We then generate a displaced camera pose $\mathbf{r}$ that is directed towards another scene point $\mathbf{X}$, located close to $\mathbf{X}^*$. The further $\mathbf{X}$ is from $\mathbf{X}^*$, the less overlap there will be between the images associated to the starting pose $\mathbf{r}$ and the desired one $\mathbf{r}^*$. In Fig. 1, we examine the convergence rate of the different methods as a function of the distance between $\mathbf{X}$ and $\mathbf{X}^*$. The figure illustrates, that even when there is a low distance between the points (and thus a high overlap), dimensionality reduction approaches (especially AEVS and PCA [2]) are effective as they greatly improve convergence. At a higher distance, our method outshines the others and is more resilient to the low image overlap.

For the second scenario, where a screwing motion must be performed, we examine the performance depending on the motion to be performed. For each displacement, we run servoing for 10 different desired poses and obtain the average convergence rate. The rotation around the optical axis is in the range [-70°, 70°] and translation between -30cm (starting camera closer to the scene) and 30cm (camera is farther to the scene than the desired one). We report the results in Fig. 2. While DVS [3] converges across a wide range of displacements, the results degrade when there is a large rotation. In these cases, AEVS fares better. It is also less sensitive to the translational motion. These results clearly support the claim of the improved convergence cone, claimed in [1].

Authors are with Univ Rennes 1, Inria, CNRS IRISA, Rennes, France Email: {samuel.felton, pascal.brault, elisa.fromont, eric.marchand}@irisa.fr. Elisa Fromont is also with Institut Universitaire de France.

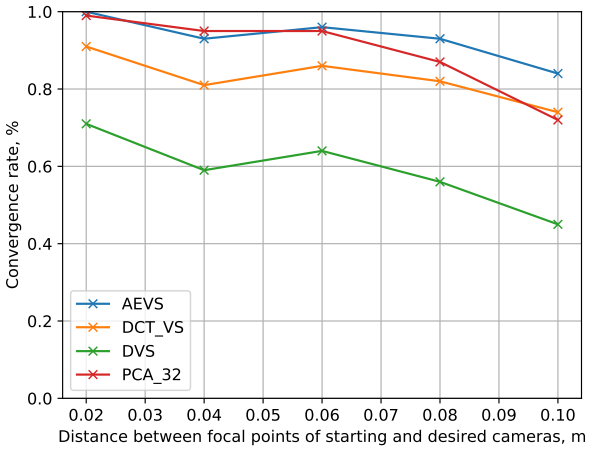
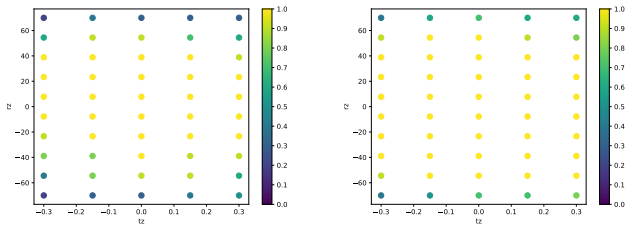


Fig. 1: Convergence rate as a factor of the distance between the focal points (in 3D space) of the starting and desired views. This metric directly correlates with overlap between the two images.



(a) DVS (b) AEVS

Fig. 2: Screw motion test case: rate of convergence given the translational displacement on the z axis (in meters) and the rotation around the z axis (in degrees). The results are averaged over 10 different desired poses and the color indicates the convergence rate of the methods for a given displacement.

II. Ablation study

In this section, we study how our method scales with the number of parameters of the network, as well as the dataset size used for training. This is a complement to the original ablation study, in Section IV.b of the paper [1]. We evaluate these in a single scene setup, on the Hollywood triangle scene, shown in Figure 7 of the original paper.

To evaluate the impact of the size of the encoder, we run our tests in simulation (similar to the ones reported in section IV-B of [1]) on models with various sizes. The encoder used in the base paper is a ResNet-18 [4]. To see whether smaller models may work, we reduce the number of filters in the convolutional blocks. We also examine the performance of larger models,

we train ResNets with 34 learnable layers. For each model, we perform 5 runs, and report their average convergence rate, as well the standard deviation.

Encoder	# of parameters	Look-at test	Screw motion test
ResNet-18, 25% of filters	0.75M	88.0 ± 5.6	82.6 ± 2.06
ResNet-18, 50% of filters	3M	92.08 ± 1.1	86.16 ± 1.75
ResNet-18 [1]	11.99M	92.56 ± 0.8	88.84 ± 1.5
ResNet-34	22.99M	92.28 ± 0.54	86.48 ± 1.0

TABLE I: Simulated servoing results with models of different sizes.

The results in Table I highlight that the performance is in part tied to the network size. It is especially apparent, that as the number of parameters grows, the variance of the results diminishes. While the ResNet-18 with 25% of filters converged successfully and reached similar loss values, the servoing results were heterogeneous. The ResNet-34, while a bigger model, obtained results close to those of the ResNet-18.

Next, we investigate the impact of the number of samples in the learning stage. To do so, we train our autoencoder on datasets with 200, 2k and 100k images. The image generation process is the same. We keep the number of iterations fixed, so that all models have the same amount of training. We use the same number of optimization steps as in the experiments of the original paper, with a batch size of 50. The networks are trained for 20k iterations. For each sample count, we train 3 models on different datasets with the same number of images. We report the average convergence rate as well as the accuracy in the cases where they converge.

Dataset size	Average val error	Look-at test		Screw motion test	
		Convergence %	End error mm, °	Convergence %	End error mm, °
200	0.667	27 ± 8.53	2.14, 0.2	84.4 ± 2.65	1.34, 0.127
2k	0.507	86 ± 3.3	0.097, 0.009	89.26 ± 0.1	0.18, 0.017
100k	0.348	91.46 ± 1.5	0.029, 0.002	86.7 ± 0.8	0.066, 0.006

TABLE II: Simulated servoing results with datasets of different sizes. The first column details the average loss value of the trained network on the validation set. The latter columns show the convergence rate and end positioning error for the cases that converged on the two different test cases that converged

Table II shows that the dataset size (and overfitting) seems to have a strong impact on the servoing performance in the look-at test. When only 200 samples are available, the performance is worse than that of DVS, that converges in 59% of the cases. As the size of the dataset grows, the convergence rate improves. Another visible general trend is that the accuracy of the servoing improves as the dataset size grows. Indeed, while networks trained with 200 samples have an average end positioning error in the order of 1-2 millimetres and 1-2 tenth of a degree, those trained with 100k images are far more accurate, with translation errors below a tenth of a millimetre and a hundredth of a degree. The results are less clear for the screw motion test, where the 2k models perform better than the 100k networks.

III. Simulated multiscene experiment

To evaluate the performance of our method in a multiscene setting, we train a network on the ImageWoof dataset[1], a subset of ImageNet [5] containing only images of dogs of different breeds. The dataset has 9k training images, and close to 4k validation images, of which we reserve 200 for testing. Some of the testing images (from which viewpoints will be generated) are displayed in Fig 3. The images are varied, with some having low texture (uniform background such as the sky or sea), while others have high frequency patterns.

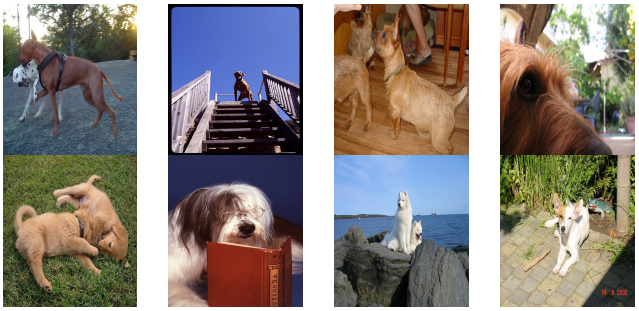


Fig. 3: Examples of ImageWoof testing scenes.

The encoder is a ResNet-34, with the decoder being scaled accordingly, as a smaller network did not converge to a desirable solution. The latent dimension is set to 64, to allow for the capture of more information in the bottleneck. We generate 100K viewpoints, spread uniformly across the different "scenes". For each scene, we evaluate on 20 servoing examples and, as there are 200 testing scenes, we have 4000 samples in total.

We compare the results of DVS [3] and AEVS for these scenes and display them in Table III. For the look-at test, the difference in convergence is not as strong as when AEVS is trained on a single image, but is still an improvement over DVS. The difference is more pronounced on the screw motion, where using AEVS results in a 13 point improvement.

	Look-at test	Screw motion test
DVS [3]	70.6	73.65
AEVS	76.4	87.25

TABLE III: Convergence rate on unseen scenes of the ImageWoof dataset.

These results highlight the fact that our approach is still of interest when considering a large number of scenes.

IV. Robot experiments

In the following section, we try our method on different scenes and show the results. The first scene is an electronic component that features small and repetitive patterns. We position the camera 30cm above the board. The scene features small depth disparities. To perform our experiment, we first train a network on image of the component, in the same fashion as done in the original paper. To account for the higher frequencies, we use a larger latent space and set the dimension of $\mathbf{z}$ to 64. For our first example, we start with an initial error $\Delta\mathbf{r}_0$ = (-13.7cm, 16.29cm, -5.95cm, 12.7°, 17.01°, -14.15°). The scene only contains the electronic component. Fig. 4 shows the overall results of our method. By minimising the error in the latent space (Fig. 4g), the error in the image space, visible in Fig. 4d is correctly reduced. Similarly, the

[1]Available at https://github.com/fastai/imagenette

pose difference, shown in Fig. 4e is correctly minimised, with the final error being $\Delta\mathbf{r}_{final}$ = (0.02cm, -0.22cm, -0.02cm, -0.37°, -0.03°, 0.02°).

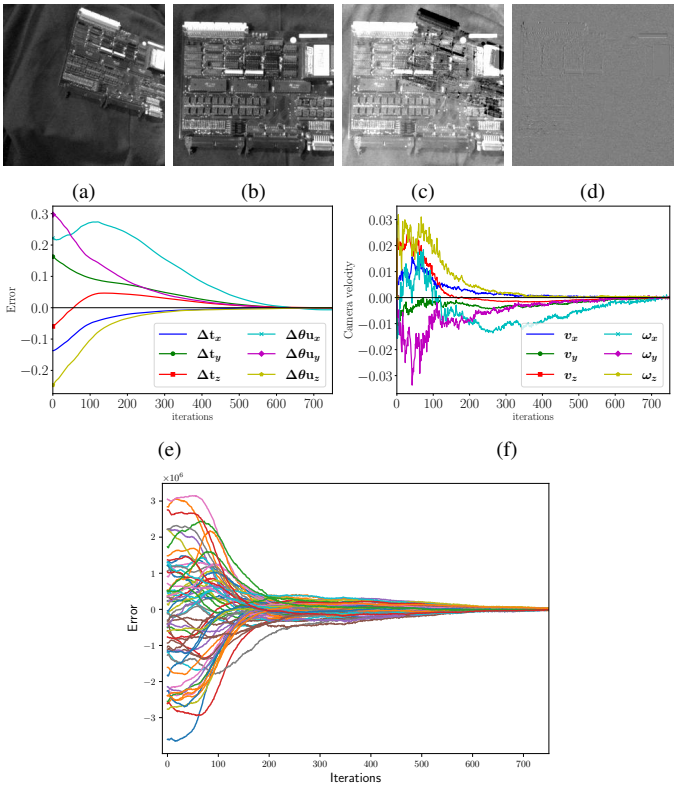


Fig. 4: Positioning task wrt. an electronic board: (a) Starting image $\mathbf{I}$. (b) Desired image $\mathbf{I}^*$. (c) Starting image difference $\mathbf{I} - \mathbf{I}^*$. (d) Final image difference. (e) Positioning errors (in m and rad). (f) Camera velocities $\mathbf{v}$. (g) Error in the latent space $\mathbf{z} - \mathbf{z}^*$.

In the next example, displayed in Fig. 5 we add distractors to the scene in order to hide the more distinguishable parts of the board. The initial pose difference is $\Delta\mathbf{r}_0$ = (-9.85cm, 10.17cm, -4.96cm, 16.05°, 8.16°, 14.87°). For this experiment, the convergence was slower and the motion was less straightforward. Nevertheless, the pose error is greatly reduced, reaching a final error $\Delta\mathbf{r}_{final}$ = (0.02cm, 0.2cm, 0.06cm, 0.36°, -0.02°, -0.15°).

For the next experiment, shown in 6, our scene is an industrial component (hydraulic connector). This scene is hard for direct methods, as it contains very little information: most of the image is white and has very little gradient. It also features strong shadows. Because of this, it can be easy to fall into local optimums and for the servoing with DVS [3] to not reach the desired pose. For the demonstration of Fig. 6, our initial error is $\Delta\mathbf{r}_0$ = (-44.22cm, 7.23cm, -3.13cm, 7.61°, 30.19°, -4.83°) and the image at the starting pose is displayed in Fig. 6a. The end image, visible in Fig. 6b, is a closer side view of the component and due to the camera orientation is not planar. While DVS and DCT-based VS [6] fail in this case, our method successfully controls the robot and greatly reduces the pose error, as seen in Fig. 6e. The final error is $\Delta\mathbf{r}_{final}$ = (-0.05cm, -0.0cm, -0.21cm, 0.05°, -0.01°, -0.05°).

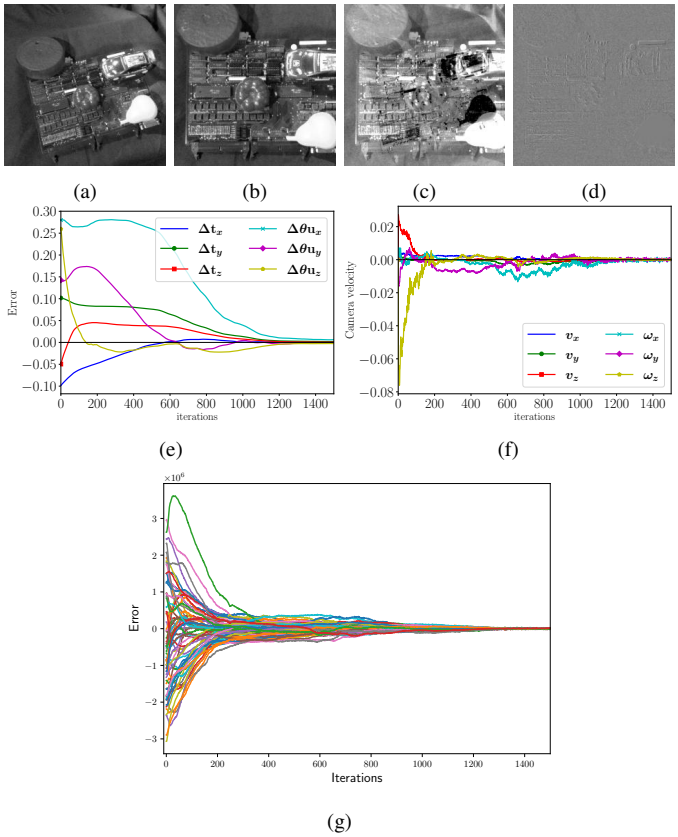


Fig. 5: Positioning task wrt. an electronic board featuring occlusions (a) Starting image $\mathbf{I}$. (b) Desired image $\mathbf{I}^*$. (c) Starting image difference $\mathbf{I} - \mathbf{I}^*$. (d) Final image difference. (e) Positioning errors (in m and rad). (f) Camera velocities $\mathbf{v}$. (g) Error in the latent space $\mathbf{z} - \mathbf{z}^*$.

Finally, in the last experiment, we reuse the network trained for the multiscene experiment in Section III. At the center of our scene, we position a small dog figurine, along with other small objects. This scene contains large depth disparities, as the top part of the desired image (Fig. 7b) is 40cm away from the camera, while the majority of the scene is 30cm away. The dog has a width of 10cm, further reinforcing this disparity. In this example, the servoing must bring the dog into the center of the image, and must realise a strong forward motion. The motion on the y axis is also large, and is not compensated by a rotation around the x axis, making the error in the image space (Fig. 7c) stronger. Starting from a pose error of $\Delta\mathbf{r}_0$ = (8.5cm, -17.58cm, -14.78cm, -6.92°, -8.68°, -5.41°), we minimise the error in the latent space, of the multiscene network trained on ImageWoof (displayed in Fig. 7g) and successfully converge close to the desired pose, with a very low final image error (Fig. 7d). While the convergence is not as smooth in the 3D space, the decrease in the image error, visible in Fig **??**) is fairly constant. The final pose error is $\Delta\mathbf{r}_{final}$ = (0.23cm, -0.07cm, 0.06cm, -0.13°, -0.4°, 0.1°).

REFERENCES

[1] S. Felton, P. Brault, E. Fromont, and E. Marchand, "Visual servoing in autoencoder latent space." Submitted, 2021.

[2] E. Marchand, "Subspace-based visual servoing," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2699–2706, July 2019.

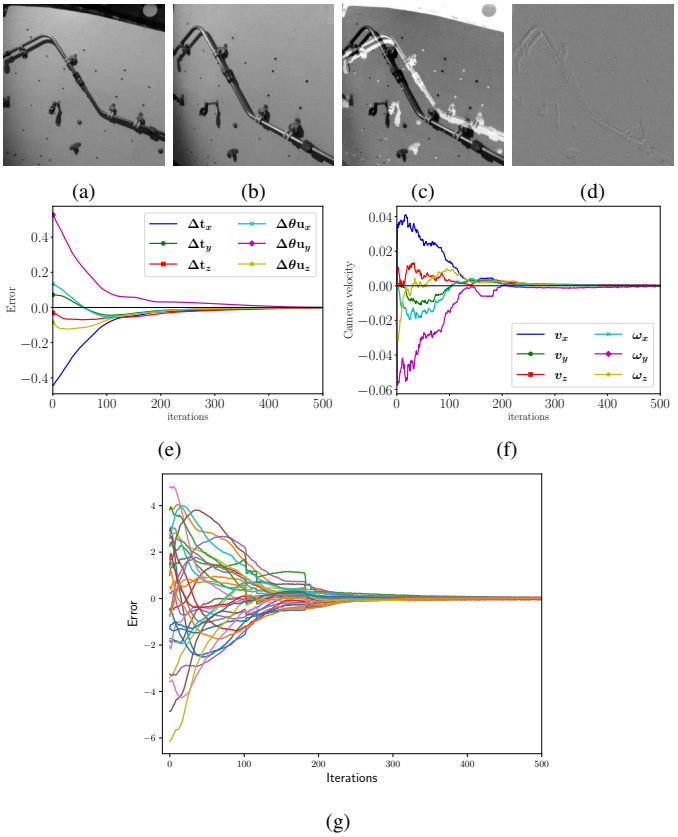


Fig. 6: Positioning task wrt. an hydraulic connector (a) Starting image $\mathbf{I}$. (b) Desired image $\mathbf{I}^*$. (c) Starting image difference $\mathbf{I}-\mathbf{I}^*$. (d) Final image difference. (e) Positioning errors (in m and rad). (f) Camera velocities $\mathbf{v}$. (g) Error in the latent space $\mathbf{z}-\mathbf{z}^*$.

[3] C. Collewet, E. Marchand, and F. Chaumette, "Visual servoing set free from image processing," in *IEEE Int. Conf. on Robotics and Automation, ICRA'08*, Pasadena, CA, May 2008, pp. 81–86.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[6] E. Marchand, "Direct visual servoing in the frequency domain," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 620–627, Apr. 2020.

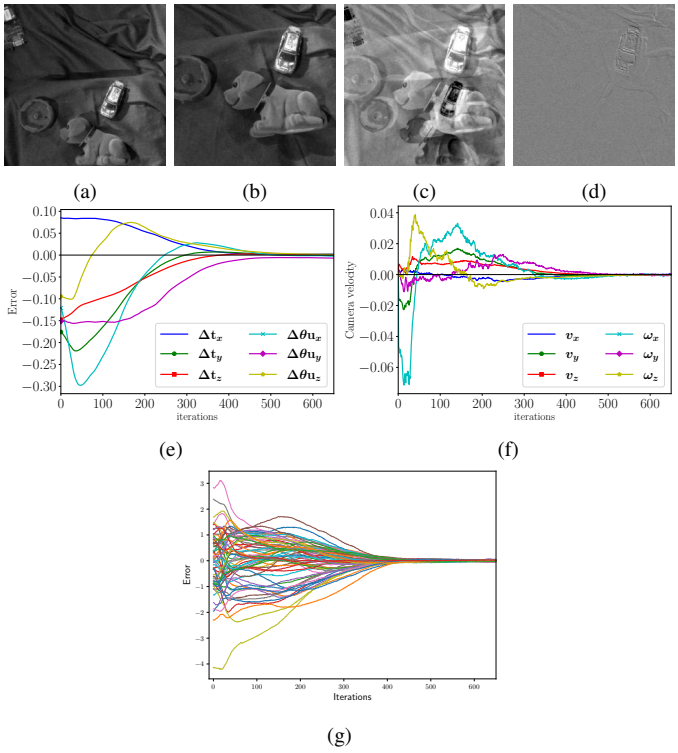


Fig. 7: Multiscene experiment (network was trained with a set of dog images), servo on a scene that features a dog (a) Starting image $\mathbf{I}$. (b) Desired image $\mathbf{I}^*$. (c) Starting image difference $\mathbf{I}-\mathbf{I}^*$. (d) Final image difference. (e) Positioning errors (in m and rad). (f) Camera velocities $\mathbf{v}$. (g) Error in the latent space $\mathbf{z}-\mathbf{z}^*$.