CEAS EuroGNC 2022

"Conference on Guidance, Navigation and Control"

3-5 May 2022 @ Technische Universität Berlin, Germany

Spectral Loss for Monocular Self-Supervised Depth and Visual Odometry in Rover Navigation

Juan Ignacio Bravo Pérez-Villar PhD Student, Deimos Space S.L.U., Flight Systems, 28760, Madrid, Spain. juan-ignacio.bravo@deimos-space.com

Álvaro García-Martín Associate Professor, Universidad Autónoma de Madrid, Video Processing and Understanding Lab, 28049, Madrid, Spain. alvaro.garcia@uam.es

Jesús Bescós Associate Professor, Universidad Autónoma de Madrid, Video Processing and Understanding Lab, 28049, Madrid, Spain. j.bescos@uam.es

ABSTRACT

This article explores the use of monocular self-supervised visual odometry and depth estimation algorithms in rover-like scenarios. The aim of these methods is to learn, using Convolutional Neural Network (CNN) architectures, the estimation of pixel-dense depth maps and visual odometry measurements from monocular video sequences without any associated ground-truth data. The article reviews the core method, its limitations, and the solutions proposed in the literature. Different learning objectives from the literature are tested and the associated results are reported. In addition, a new learning objective based on the frequency domain of the image is proposed to exploit the particularities of the rover-like scenarios, increasing the accuracy of the odometry results.

Keywords: Self-Supervised; Visual Odometry; Monocular; Depth; DCT

Nomenclature

D	=	depth
DCT	=	Discrete Cosine Transform
I	=	image
K	=	camera intrinsics
p	=	pixel coordinates
R	=	rotation
$SSIM$	=	Structural Similarity Index Metric
T	=	transformation matrix
t	=	translation

1 Introduction

Visual odometry and depth estimation are of key importance in robot-based autonomous navigation. Recent advances in computer vision allow to estimate dense depth maps from monocular images, relaxing the need for camera stereo pairs or power-demanding LIDAR equipment. Recent methods used to require ground-truth depth data for training. However the availability of relevant training data in space applications appears as a key issue, as complete training data is only available in simulated or mock-up environments yielding to new problems of domain gap and generalisation.

The recent introduction of self-supervised monocular depth and visual odometry algorithms mitigate the constraint on ground-truth data availability. These methods can be trained purely on monocular image sequences, without the need of ground-truth depth and/or visual odometry poses. This appears as an opportunity for rover-like exploration missions, where the captured data by the rover can be used to incrementally train and and improve visual odometry and monocular depth sensing capabilities.

In this article we explore the use of such methods in a rover-like dataset, performing a comparison among different methods and show which techniques are extensible to rover-like scenarios. In addition, we propose a spectral-based loss for image reconstruction, based on previous work of our research lab [*article pending publication, currently under review]*, that captures frequency information for the image reconstruction process. The paper is organised as follows: The related work is presented in Section 2. The method is described in Section 3. The dataset, experiments and discussions are presented in Section 4. Finally the conclusions are presented in Section 5.

2 Related Work

Self-supervised monocular depth and pose estimation models are trained to reconstruct a target frame from a sequence of nearby views using the estimated relative camera pose and depth [1]. The photometric difference between the reconstructed frame and the target frame is used as a supervisory signal to improve the predictions (see Fig. 1). However, such process implicitly assumes several key elements 1) the scene is static, i.e. without moving objects that cause occlusions not explainable by the camera motion; 2) the scale between the predicted depth and pose are consistent during training; 3) test and the camera intrinsics remain the same at train and test time.

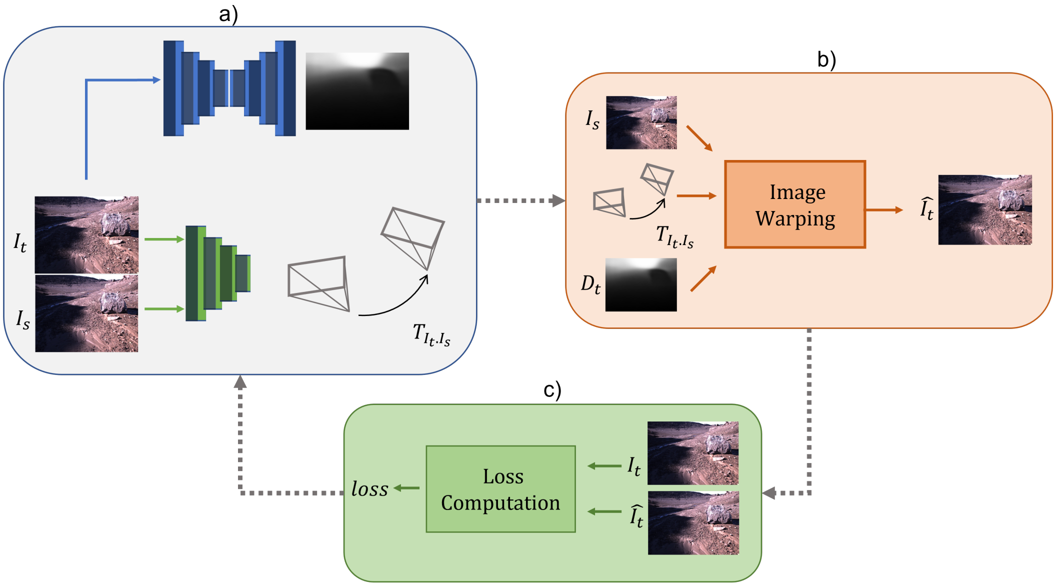


Fig. 1 Self-supervised visual odometry and depth estimation pipeline. a) the relative camera pose is estimated between a pair of frames, a source frame I_s and a target frame I_t using a encoder-like architecture (green); a encoder-decoder (blue) is used to estimate the depth of the target frame I_t; b) an estimation of the target frame $\hat{I}_t$ can be computed from the source frame I_s using the estimated target depth D_t and the relative pose between I_t and I_t. c) the differences between I_t and $\hat{I}_t$ define the learning objective

2.1 Reconstruction Assumptions

The assumption on static scenes and the absence of occlusions are hardly met in real-world scenes, particularly in road and city environments where many objects move at different speeds and trajectories with respect to the camera. Also, surfaces with high reflectance might also appear in real-world objects creating inconsistent photometric errors. The common approach is to exclude from the loss computation the pixel regions that cannot be explained by the ego-motion movement. One of the first self-supervised monocular visual odometry and depth estimation methods, [1], proposed a combined pose-explainability network to retrieve both the camera pose between frames and a mask that provided information on which pixels were not explainable by the predicted motion. Other authors [2–4] explicitly model the scene motion with optical flow to mask areas of the image that violate the ego-motion assumption by measuring the forward and backward optical flow consistency between two frames. [5, 6] classify scene objects as "possibly moving" or "static" by injecting semantic information and then mask moving areas in the computation of the loss. An occlusion-aware loss term based on measuring depth differences for corresponding points was proposed in [5] to model the occlusions directly at the loss level. The authors of [7] use the minimum per-pixel error inside a training sequence to mitigate the influence of occlusions, and use the warped source frame for the loss computation in those areas where the error between the warped source frame and the target frame is lower than the error between the source frame and target frame. [8] computes a mask based on the distribution of the image reconstruction loss, and introduces weak supervision based on epipolar geometry estimations. Other authors directly use the depth to generate 3D point clouds and create depth alignments for additional supervision [9, 10].

2.2 Scale Assumptions

In a monocular setting, if no other cues are available, the predicted depth and pose are scale-ambiguous. The sensed objects may be large objects located far away from the camera, or small objects that are close to it. To enforce scale consistency during training, several loss terms have been proposed. Geometric consistency between estimated depths at source and target frames is enforced in [11] by minimising a loss based on the difference between the source depth warped to the target frame and the target depth. A similar approach is followed by [12] but using the inverse depth terms. Other authors explicitly disentangle scale from the problem by aligning the depth estimations to sparse depth points obtained from triangulation as presented in [13].

In a rover scenario, scale can be estimated by integrating other sources of data. Prior knowledge on the height of the camera can be leveraged to estimate the scale of the translation vector during the visual odometry process [14], although such techniques usually assume a flat and homogeneous ground plane. Other data sources like wheel odometry estimates or sparse point LIDAR depth measurements can be used to reconstruct up to scale visual odometry and depth map estimates.

2.3 Camera Assumptions

Respect to the intrinsic camera parameters, the generalisation of camera pose estimation to tests sets that differ from the training sets is often neglected and few works exist in this direction. Explicitly adding intrinsic information to the images by incorporating a new layer in the learning architecture is treated in [15]. Estimating the rotation and translation from point correspondences using epipolar geometry instead of using a network is presented in [10, 13]. [16] uses a combination of Perspective-n-Point (PnP) and fundamental matrix decomposition to solve for cases where the total translation results in degenerate epipolar solutions.

The aim of this article is to evaluate the most representative state-of-the-art contributions on a rover-like scenario. Also, we extend previous work by proposing the use of a spectral loss in the task of image reconstruction, leveraging that in a rover-like environment almost all the pixel motion is due to the camera ego-motion (excepting shadows cast by the rover, or small moving rocks). Generalisation to different camera models is not treated here, as that would require a rover-like dataset of the same environment captured with different cameras following the same trajectory.

3 Method Description

Most of the self-supervised monocular depth and pose estimation methods available in the literature employ image reconstruction from time contiguous video frames as the supervisory signal. Image reconstruction is the task of estimating a target frame by warping past or future source frames under the assumption of a given pixel or camera movement. The image reconstruction task is modelled as a transformation of the camera coordinate system, so that the differences between the target and reconstructed frames can be used as a supervision signal to learn relative camera poses and depth.

The relationship between the pixel coordinates p_t of a target frame I_t and the corresponding pixel coordinates p_s of a source frame I_s can be related by means of the scene depth and the camera motion via Eq. 1. Let K denote the camera intrinsics matrix, $T_{t->s}$ the transformation matrix between the target and source camera coordinates and D_t the depth of the target scene. For a given pair of frames, the pixel coordinates of a target frame I_t can be transformed to those of a source frame I_s by first retroprojecting the target pixel homogeneous coordinates into the 3D world coordinates, then rotating and translating the resulting point cloud to the source coordinate system, and finally projecting the point cloud to source frame coordinates. The values of the pixels in projected coordinates p_s are used to obtain the reconstructed target frame $\hat{I}_t$, i.e. the source frame warped to the target pixel coordinates.

$$p_s = KT_{t->s}D_t(p_t)K^{-1}p_t \tag{1}$$

3.1 Photometric Supervision

The learning objective is then provided by the minimisation of a distance metric between the target video frame I_t and the reconstructed target frame $\hat{I}_t$. Often, distance metrics are based on the differences between the target and the reconstructed frame in the pixel domain, under the assumption that the surfaces are Lambertian, i.e. the apparent intensity is constant independently of observation angle. Two common metrics used used in the literature [1, 2, 11] are the L1 pixel intensity distance and the Structural Similarity Index Metric (SSIM).

The L1 pixel intensity distance is expressed as cumulative sum of absolute pixel-value differences between the target and reconstructed images:

$$L_1 = \sum_p |I_t(p) - \hat{I}_t(p)| \tag{2}$$

SSIM is a quality measure presented by [17] based on the comparison of luminance $l(a,b)$, contrast $c(a,b)$, and structure $s(a,b)$, between two images a and b using sliding circular-symmetric Gaussian windows of NxN pixels. The luminance measure is expressed as $l(a,b) = \frac{2\mu_a\mu_b+C}{2\mu_a^2\mu_b^2+C}$, where μ_a and μ_b represent the mean intensity values of the a and b images inside the window and C represents a numeric constant to avoid numerical instability. The contrast measure is computed as $c(a,b) = (2\sigma_a\sigma_b + C)/(\sigma_a^2 + \sigma_b^2 + C)$, where σ stands for the standard deviation of the mean centred image. The structure index $s(a,b) = (\sigma_{ab} + C)/(\sigma_a\sigma_b + C)$ is computed using the correlation coefficient σ_{ab} computed between the mean centred and normalised standard deviated images.

$$SSIM(a,b) = \frac{(2\mu_a\mu_b + C)(2\sigma_{ab} + C)}{(\mu_a^2 + \mu_b^2 + C)(\sigma_a^2 + \sigma_b^2 + C)} \tag{3}$$

The SSIM loss term is then computed as:

$$L_{ssim} = \frac{1 - SSIM(I_t, \hat{I}_t)}{2} \tag{4}$$

3.2 DCT Supervision

Phometric based losses arise naturally in the direct comparison of two images and allow to easily exclude from the learning objective image areas that are not explainable by camera motion (e.g. a moving car), improving the learning objective in these situations [5, 7]. However, in a rover-like environment it is expected that all the observed motion is due to the camera motion. This particularity allows to explore transformed domains in which more optimal image representations for the learning objective may be achieved.

A loss term based on the Discrete Cosine Transform (DCT) is proposed to account for frequency information as a similarity measure between the target and reconstructed images. The DCT is a real valued transform that expresses a signal as a linear combination of cosine bases that represent the frequency components (pixel-intensity variability) across rows, columns and linear combinations of them. The DCT has been widely used to characterise local structures in image signals and image compression. Eq. 5 expresses the DCT of an image, $x(n,m)$, of size NxM.

$$DCT(X) = X(u,v) = C_u C_v \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n,m) cos(\frac{2n+1u\pi}{2N}) cos(\frac{2m+1v\pi}{2M}) \tag{5}$$

$C_u = \sqrt{1/N}$ when $u = 0$, else $C_u = \sqrt{2/N}$. $C_v = \sqrt{1/M}$ when $v = 0$, else $C_v = \sqrt{2/M}$. Then, the DCT based loss is computed as:

$$L_{DCT} = \|DCT(I_t) - DCT(\hat{I}_t)\|_2 \tag{6}$$

In the computation of the L_{DCT} the first coefficient of the DCT is removed, as it represents the mean value of the image.

3.3 Weak Pose Supervision, Smoothness Loss, and Scale Consistency

The problem presented in Eq. 1 is an ill-posed problem. There are infinite combinations of depth values that may suffice the pose objective. Two solutions are generally introduced to overcome this issue. Smoothing loss and weak pose supervision

The smoothness loss enforces smooth depth maps in areas where the magnitude of the gradient in the image is low. This term, known as edge-aware smoothness loss is computed as:

$$L_{smooth} = |\partial_x d_t^*| e^{-|\partial_x I_t|} + |\partial_y d_t^*| e^{-|\partial_y I_t|}, \tag{7}$$

where ∂_x and ∂_y indicate derivatives in the horizontal and vertical direction. Following [18] we use d_t^*, the mean normalised inverse depth, that penalises the shrinking of the estimated depth into small values.

To penalise pose errors that might yield into incorrect depth and pose estimations, we introduce a weak pose supervision loss based on the essential matrix decomposition, in a similar fashion to [8]. The essential matrix for a given pair of images is computed using key-point matches between them, obtained via a differentiable implementation of a scale space detector with Multiple Kernel local descriptors [19, 20]. The essential matrix is then decomposed into rotation and translation (R_{ess}, t_{ess}) and the optimal solution is chosen as the rotation and translation (R_{ess}, t_{ess}) that maximises the number of triangulated key-points in front of the camera plane. We compute the pose loss term as:

$$L_{pose} = \|I - R_{ess}R^T\|_{fro} + \|\bar{t}_{ess} - \bar{t}\|_2, \tag{8}$$

where the sub-index *fro* denotes the Frobenius norm. The translation vectors are normalised before comparison to remove the effect of the scale in the calculation

Scale consistency between the estimated depth and motion is often neglected and a scale-consistent output is learnt by the network. In this setting, the ego-motion network cannot provide a full camera trajectory over a long video sequence because of the per-frame scale ambiguity [11]. To explicitly enforce the motion and the depth to provide a scale consistent output, we use the geometric consistency loss from [11]:

$$L_{scale} = \sum_{p} \frac{|\hat{D}_t(p) - D_t(p)|}{\hat{D}_t(p) + D_t(p)} \tag{9}$$

3.4 Learning Objectives

To estimate the influence of each loss term in the performance we define different learning objectives. We present a summary of each loss term and their associated weights in Tab. 1

Name	Loss terms	Values
Baseline	L_1 + γL_{smooth}	$\gamma = 0.001$
Min Reprojection	L_1 + γL_{smooth}	$\gamma = 0.001$
SSIM	αL_1 + $(1-\alpha)L_{ssim}$ + γL_{smooth}	$\gamma = 0.001, \alpha = 0.95$
DCT	L_1 + αL_{DCT} + γL_{smooth}	$\gamma = 0.001, \alpha = 0.01$
Pose	L_1 + αL_{pose} + γL_{smooth}	$\gamma = 0.001, \alpha = 0.001$
Scale	L_1 + αL_{scale} + γL_{smooth}	$\gamma = 0.001, \alpha = 0.5$

Table 1 Different learning objectives defined for the experiments with their associated weights for each term.

The weight of each loss term has been chosen according to the reported values literature, excepting the value of the SSIM whose typical value is $\alpha = 0.15$. In our experiments such value made the results diverge quickly leading to large errors in pose and depth, we tested gradually low values of α until the learning objective converged. The Min Reprojection experiment has the same learning objective as the Baseline, but it is designed to measure the influence of excluding areas from the computation loss in a rover-like scenario, where the assumption of motion being different to the ego-motion is not broken. To do so, the approach from [7] is followed: for a given sequence only the minimum error across frames is kept, and the pixels from the reconstructed frame $\hat{I}_t$ that give lower loss values than those that use directly the source frame I_s are used.

3.5 Architecture and Other Considerations

The architecture is based on the approach from [7]. The depth is estimated through a CNN encoder-decoder architecture with skip connections. The encoder is a ResNet 18 [21] pretrained on ImageNet [22]. The decoder outputs the depth maps at different levels of the decoder (scales) and uses ReLu activation functions except for the depth decoder head that uses scaled sigmoids to output the depth in a fixed, controlled range. Following other works we fix the output to 0.1-100 units of depth.

The pose is also estimated using a CNN-based architecture. We follow [7] by using a modified Resnet-18 that accepts two images concatenated in the channel dimension. The network outputs the rotation and translation of the images in axis angle representation. Following other works, we make use of the locally sub-differentiable bi-linear sampling to sample the adjacent views. To overcome the gradient locality of the bi-linear sampling, we follow the multi-scale approach of [7]. We use the depth maps extracted at different levels of the depth estimation network to reconstruct the image, however, instead of computing the photometric error loss at different resolutions, the depth is scaled at the resolution of the image. In this way all the layers work towards the same objective.

4 Dataset and Experiments

We train and evaluate each system on the Devon Island Rover Navigation dataset [23]. This dataset provides rover traverse data and long-range localisation data captured over 10 km in vegetation-free, planetary-analogue terrain. We use the left camera and the RTK differential GPS of the rover traverse data to evaluate our system. From the 22 sequences of images, totalling 49,410 image pairs, we train our experiments with the images acquired with the RGB left camera of the first 4 sequences (a total of 10,028 images) and use the sequences 04, 05, 06 and 07 for evaluating the algorithm outside the training set. The images are resized to 408x640 px from the original 1280x960 px and no data augmentation is used. Each batch is composed of image triplets with the preceding and following frames to the target frame.

The experiments are implemented in PyTorch [24] using the Adam optimiser [25] with a learning rate of 10^{-4}. We make use of pre-trained ResNet-18 backbones for the depth and pose estimation networks. We also make use of the Kornia library [20] for differentiable image warping and key-point matching.

4.1 Visual Odometry Performance

The influence of each loss term in the visual odometry performance is evaluated by means of the Root Mean Squared Error (RMSE) of the absolute difference between the estimated trajectory and the ground-truth. The results, reported numerically in Tab. 2 and graphically in the Appendix, are obtained for both the training and test sequences and computed over each individual sequence. The super-index * indicates the sequence was used for training.

Since the relative motion estimated by visual odometry systems has undefined scale and orientation with respect to the ground-truth, the direct comparison of trajectories would not provide meaningful results. This issue is commonly addressed in the literature by performing a single similarity (scale, rotation and translation) alignment [26] between the ground-truth and the estimated trajectory. This alignment involves a global rotation, translation and scaling. Translating the estimated trajectory to the ground-truth instead of aligning both to share the same origin of coordinates may reduce the influence of long-term error accumulation, hence in this work the alignment is provided by a single rotation, translation to the same origin of coordinates, and a per-frame scaling. This scaling, where every translation vector is normalised and then scaled with the module of the ground-truth translation vector extracted from the GPS, is performed due to its similarity to on-line visual odometry system where every measurement is weighted with other data sources such as the Inertial Measurement Unit (IMU). The chosen alignment is designed to replicate a real operational scenario.

The results reported in Tab. 2 show that the inclusion of additional information sources in the photometric reconstruction loss (SSIM, DCT and, Min Reprojection) help to improve the Baseline learning

Method	00*	01*	02*	03*	04	05	06	07	Average Train	Average Test
Total meters [m]										
	512.69	390.17	788.50	699.44	502.99	711.26	711.64	546.21		
RMSE Error [m]										
Baseline	26.1358	11.3079	13.2032	34.8430	13.2013	15.7199	12.9380	29.7934	21.3725	17.9132
Min Reprojection	43.0327	**1.7727**	14.2009	**16.1470**	18.3547	17.0702	**10.2517**	**11.7243**	18.7883	14.3502
SSIM	26.0177	7.6808	17.4690	74.1475	**7.6474**	9.0329	17.9394	16.9418	31.3288	12.8904
Pose	44.2576	7.2900	45.3156	18.1853	38.8852	58.4711	16.4289	25.8335	28.7621	34.9047
Scale	42.8631	6.3628	26.7926	23.1558	23.9089	26.4946	20.6786	23.8841	24.7936	23.7415
DCT (Proposed)	**25.0735**	7.3755	**12.4923**	20.8762	8.5359	**8.5642**	11.5351	17.1784	**16.4544**	**11.4534**

Table 2 Error metrics for the visual odometry algorithm over different sequences. The super-index $*$ denotes the sequence was used for training. Bold-style text indicates minimum value in a column.

objective. The proposed spectral-loss for image reconstruction outperforms, in average, the existing methods. However, there is not a technique that consistently outperforms the rest in every scenario, suggesting that a combination of techniques might provide the best performance. Surprisingly, the Pose and Scale methods yield to bad results on all the scenarios. A first analysis suggests that the results of the pose objective might be caused by excessive influence of the rotation. This has been tested by smoothing the rotations during the trajectory reconstruction, by converting the rotation matrix to a quaternion representation and applying a spherical linear interpolation with the identity quaternion $q = (1,0,0,0)$. The results after this interpolation (see Tab. 1) improve, suggesting that the rotation and translation term should be weighted differently or other more suitable metrics for the rotation loss computation should be explored. The scale term enforces consistency between the depth and translation, helping to regularise the output when the displacement between frames of the vehicle is not constant or when sequences from different datasets are combined. In this dataset the motion scale is constant, having roughly 20cm of displacement between frames [23]. This might cause the learning objective try to correct for an in-existent behaviour, leading to incorrect pose and depth estimates.

Method	00*	01*	02*	03*	04	05	06	07	Average Train	Average Test
RMSE Error [m]										
Smoothed Pose	26.167	16.653	18.172	20.398	36.751	34.050	16.463	14.134	20.348	25.349

Table 3 Results after smoothing the rotation matrices for the Pose experiments by applying a spherical linear interpolation with the unit quaternion.

The depth results are only shown qualitatively due to the absence of ground-truth dense-depth maps in the dataset. A selection of relevant examples are shown in Fig. 2. The first two columns correspond to training data and the two last to test data. From the second top row to the last row, the methods shown are: Baseline, Min Reprojection, SSIM, DCT, Pose, and Scale. It can be observed how the methods fail to generalise to the presence of large rocks or other elements not seen during training. This behaviour could be explained by the lack of those elements in the training data, being almost all the sequences composed of plain terrain images. The Min Reprojection method provides depth maps with lesser detail compared to other methods. This might be caused by the masking of image-areas that cause large errors in the photometric loss. As only few examples are observed, these might be considered outliers and then rejected by the learning objective, exposing the algorithm to even lesser examples of such elements. The SSIM and DCT method provide more sharp depth maps that other losses. It is interesting to observe that the Scale method provides wrong depth measurements in the presence of large illumination gradients.

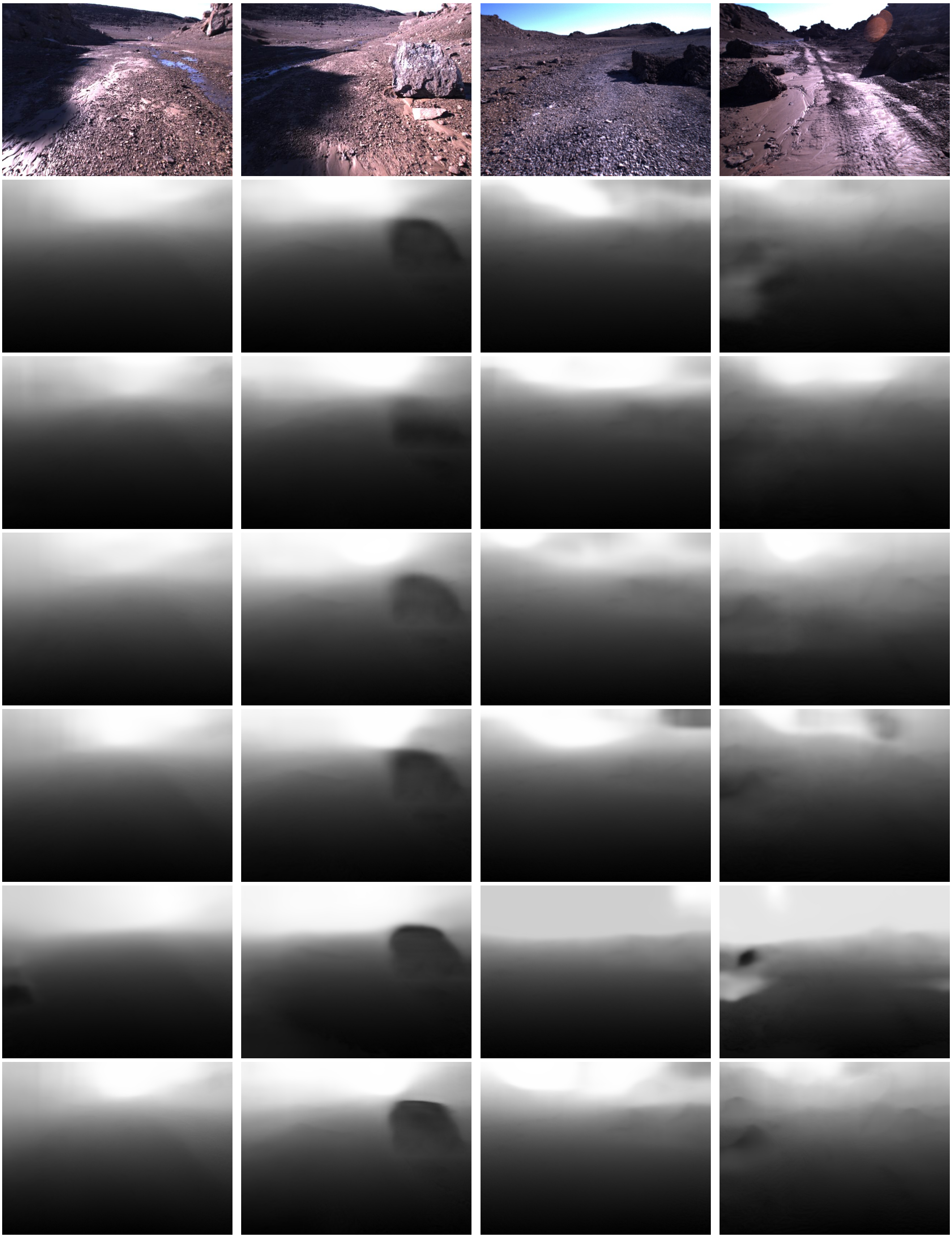


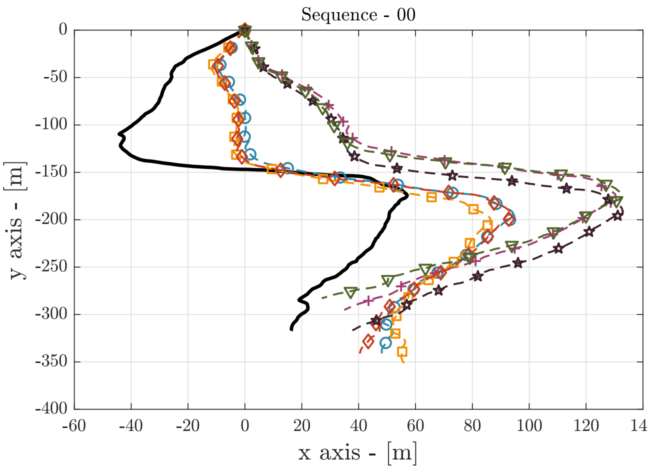
Fig. 2 Qualitative depth results. The two first columns correspond to training data, the two last to test data. From second top row to last row: Baseline, Min Reprojection, SSIM, Pose, Scale, and DCT

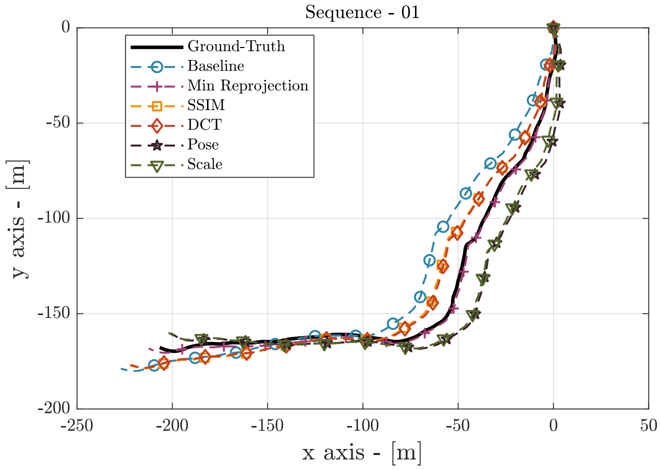
5 Conclusions

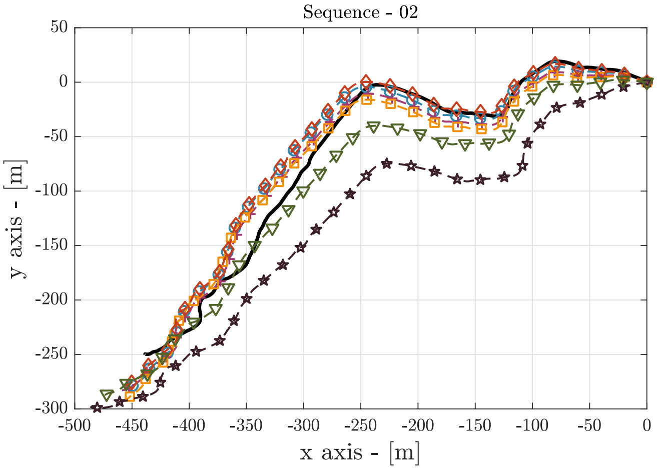
In this paper monocular self-supervised visual odometry and depth estimation methods applied to rover navigation have been presented. The majority of the state-of-the-art methods have been developed using datasets captured from moving vehicles in urban environments that present different conditions to rover-like scenarios: motion different to the ego-motion, different vanishing points, and larger variety of objects. We have shown that such methods generalise well when applied to rover-like environments. However, the use of techniques that explicitly mask areas from the reconstructed image that arise in high-error might affect the depth estimation of objects with low representation in the dataset. The fact that the ego-motion assumption is not broken in rover-like scenarios can be leveraged to introduce new transformed domain losses that include the frequency information of the image and not only the pixel photometric error, leading to better odometry measurements.

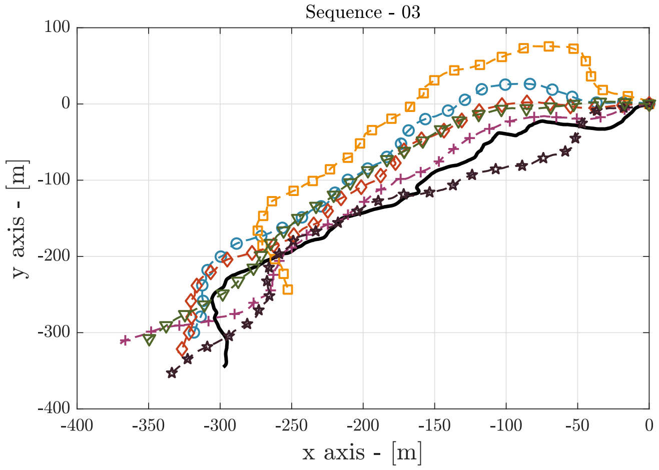
We hope the presented work motivates further research in the are of monocular self-supervised visual odometry and depth estimation in rover scenarios. Suggested future lines of research are the inclusion of other sensors to improve the estimates (e.g. introduce scale information), and study the performance of the algorithm when tested outside the training domain: different camera model, moving at a different speed, and changes in the image conditions such as rock sizes.

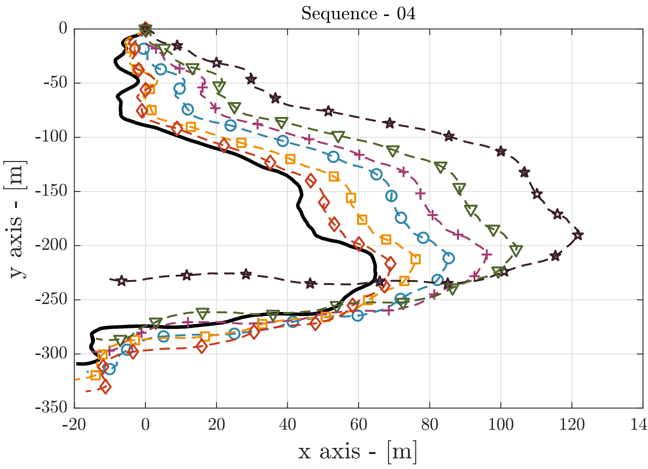
Appendix

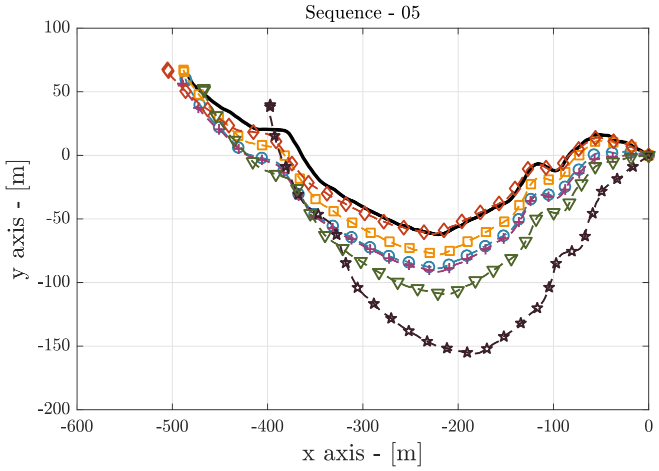

Sequence - 00
y axis - [m]
x axis - [m]
0
-50
-100
-150
-200
-250
-300
-350
-400
-60
-40
-20
0
20
40
60
80
100
120
140

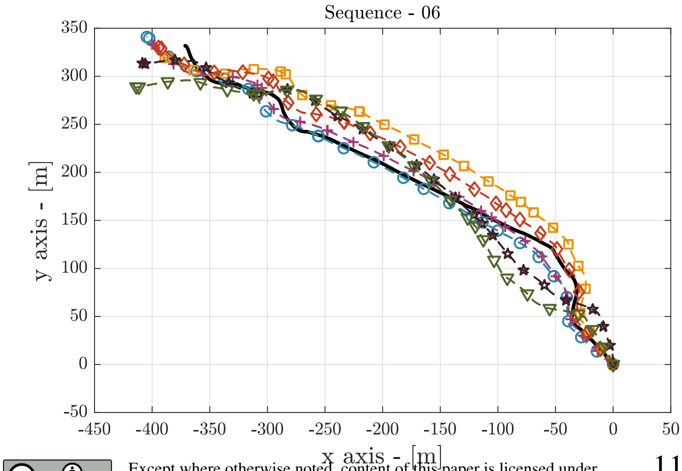

Sequence - 01
Ground-Truth
Baseline
Min Reprojection
SSIM
DCT
Pose
Scale
y axis - [m]
x axis - [m]
0
-50
-100
-150
-200
-250
-200
-150
-100
-50
0
50

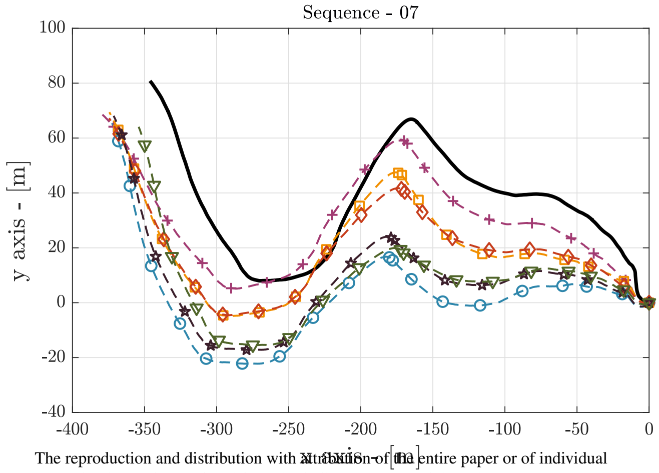

Sequence - 02
y axis - [m]
x axis - [m]
50
0
-50
-100
-150
-200
-250
-300
-500
-450
-400
-350
-300
-250
-200
-150
-100
-50
0


Sequence - 03
y axis - [m]
x axis - [m]
100
0
-100
-200
-300
-400
-400
-350
-300
-250
-200
-150
-100
-50
0


Sequence - 04
y axis - [m]
x axis - [m]
0
-50
-100
-150
-200
-250
-300
-350
-20
0
20
40
60
80
100
120
140


Sequence - 05
y axis - [m]
x axis - [m]
100
50
0
-50
-100
-150
-200
-600
-500
-400
-300
-200
-100
0


Sequence - 06
y axis - [m]
x axis - [m]
350
300
250
200
150
100
50
0
-50
-450
-400
-350
-300
-250
-200
-150
-100
-50
0
50


Sequence - 07
y axis - [m]
x axis - [m]
100
80
60
40
20
0
-20
-40
-400
-350
-300
-250
-200
-150
-100
-50
0

Acknowledgments

This research is partially funded by Consejería de Educación de la Comunidad Autónoma de Madrid, Spain.

References

[1] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.

[2] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018.

[3] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12240–12249, 2019.

[4] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. *arXiv preprint arXiv:2010.16404*, 2020.

[5] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019.

[6] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *European Conference on Computer Vision*, pages 582–600. Springer, 2020.

[7] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[8] Tianwei Shen, Zixin Luo, Lei Zhou, Hanyu Deng, Runze Zhang, Tian Fang, and Long Quan. Beyond photometric loss for self-supervised ego-motion estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6359–6365. IEEE, 2019.

[9] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018.

[10] Jiexiong Tang, Rares Ambrus, Vitor Guizilini, Sudeep Pillai, Hanme Kim, Patric Jensfelt, and Adrien Gaidon. Self-supervised 3d keypoint learning for ego-motion estimation. *arXiv preprint arXiv:1912.03426*, 2019.

[11] Jia-Wang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *arXiv preprint arXiv:1908.10553*, 2019.

[12] Huangying Zhan, Chamara Saroj Weerasekera, Ravi Garg, and Ian Reid. Self-supervised learning for single view depth and surface normal estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4811–4817. IEEE, 2019.

[13] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9151–9161, 2020.

[14] M Hossein Mirabdollah and Bärbel Mertsching. Fast techniques for monocular visual odometry. In *German Conference on Pattern Recognition*, pages 297–307. Springer, 2015.

[15] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. *arXiv preprint arXiv:2011.00359*, 2020.

[16] Huangying Zhan, Chamara Saroj Weerasekera, Jia-Wang Bian, and Ian Reid. Visual odometry revisited: What should be learnt? In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4203–4210. IEEE, 2020.

[17] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[18] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2022–2030, 2018.

[19] Arun Mukundan, Giorgos Tolias, Andrei Bursuc, Hervé Jégou, and Ondřej Chum. Understanding and improving kernel local descriptors. *International Journal of Computer Vision*, 127(11):1723–1737, 2019.

[20] D. Ponsa E. Rublee E. Riba, D. Mishkin and G. Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Winter Conference on Applications of Computer Vision*, 2020.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[23] Paul Furgale, Pat Carle, John Enright, and Timothy D Barfoot. The devon island rover navigation dataset. *The International Journal of Robotics Research*, 31(6):707–713, 2012.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[26] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(04):376–380, 1991.