Learning to listen to your ego-(motion) : Metric Motion Estimation from Auditory Signals

Letizia Marchegiani and Paul Newman

Oxford Robotics Institute, University of Oxford, Oxford, UK,
{letizia,pnewman}@robots.ox.ac.uk,
http://ori.ox.ac.uk

Abstract. This paper is about robot ego-motion estimation relying solely on acoustic sensing. By equipping a robot with microphones, we investigate the possibility of employing the noise generated by the motors and actuators of the vehicle to estimate its motion. Audio-based odometry is not affected by the scene's appearance, lighting conditions, and structure. This makes sound a compelling auxiliary source of information for ego-motion modelling in environments where more traditional methods, such as those based on visual or laser odometry, are particularly challenged. By leveraging multi-task learning and deep architectures, we provide a regression framework able to estimate the linear and the angular velocity at which the robot has been travelling. Our experimental evaluation conducted on approximately two hours of data collected with an unmanned outdoor field robot demonstrated an absolute error lower than $0.07\ m/s$ and $0.02\ rad/s$ for the linear and angular velocity, respectively. When compared to a baseline approach, making use of single-task learning scheme, our system shows an improvement of up to 26% in the ego-motion estimation.

Keywords: ego-motion estimation, sound-based odometry, deep learning, multi-task learning, acoustic sensing

1 Introduction

This paper explores the possibility of modelling a robot's ego-motion, relying only on acoustic sensing. Specifically, we employ the robot's ego-noise (i.e. the noise produced by the motors and actuators while generating motion) to make estimates on the vehicle's velocity. Optical sensors and lasers have commonly been employed to perform this task, as they are able to provide accurate pose estimates, by also overcoming the shortcomings of wheel odometry, such as error calculation during wheel slippage (e.g. [1, 2]). The performance of Visual Odometry (VO) systems are still challenged, though, by environments characterised by moderate lighting conditions or, more generally, scarcity of textures. On the other hand, laser odometry might struggle in degenerated scenes where planar areas are prevalent. Auditory perception is resilient to the scene's appearance: a property which would be particularly convenient in environments lacking structure and illumination, or in which distractions are intense. For instance, extreme scene movement could be misinterpreted as movement of the robot, and acoustic signals could be used to perform consensus checking and resolve the ambiguity. In this perspective, we

envision acoustic sensing as an auxiliary source of information, on the side of more traditional odometry methods, for the development of more robust ego-motion estimation systems. Building on such premises, in this work we introduce and evaluate a framework to estimate a robot's ego-motion, using exclusively the ego-noise produced by the vehicle and recorded by the on-board microphones. Following on from recent studies that exploit deep learning for visual and laser odometry estimation [3, 4], we model the vehicle's odometry using a deep neural network (DNN). In particular, we leverage regression analysis, and, similarly to the modelling strategy adopted by [5] for camera relocalisation, we apply a multi-task learning scheme. Unlike that work, however, we do not regress the robot's poses directly, but the linear and the angular velocity at which the robot has been travelling. We evaluate our system on approximately two hours of data collected at the University Parks in Oxford, UK.

To the best of our knowledge, this is the first work investigating robot ego-motion modelling employing solely the acoustic features of the ego-noise of the vehicle to estimate its velocity through a state-of-the-art regression framework.

2 Related Work

For long time, robot audition has mainly concerned the development of human-robot interaction frameworks (e.g. [6]). More recently, the robotics community has started investigating auditory perception in a wider perspective. A method to augment autonomous vehicles with the capability of detecting the presence of anomalous sounds (e.g. the siren of an emergency vehicle) has been introduced in [7]. Acoustic event classification in a domestic environment has been explored in [8]. In all these instances the robot's ego-noise has been examined as a limitation to the systems' performance, as introducing additional task-unrelated components to the acoustic scene, making its interpretation more challenging. Yet, robot ego-noise carries significant information which could be exploited, both for environment understanding and self-modelling. The use of ego-noise to perform terrain classification has been proposed in [9]. Sound-based self-localisation and ego-motion estimation have been approached in [10] and [11]. The former combines orientation estimates from inertial measurement unit (IMU) observations and audio-based distance estimation to localise a snake robot moving in a pipe. The latter proposes a classification framework to associate ego-noise to a set of predefined velocity profiles. Our work shares the aspirations of [11], while presenting a substantially different approach to velocity estimation. Rather than encoding the robot's motion into profiles known *a priori*, we propose a regression model able to provide, at any point in time, the current velocity of the vehicle. The resulting system will, consequently, be more flexible, and, as not relying on predefined behaviours, inherently more robust to changes in the environment leading to potential unexpected modifications in the robot's motion. Furthermore, we investigate the possibility of using auditory features to estimate the angular velocity of the vehicle, laying the foundations for the development of an audio-only ego-motion estimation system.

3 Technical Approach

Our regression analysis makes use of a deep neural network, which given as input a feature representation of the robot's motion sound, provides estimates both for the linear and angular velocity of the vehicle. A description of the features employed is presented in Section 3.3, while a more detailed illustration of the deep architecture utilised is delineated in Section 3.2.

3.1 Preliminaries

Our framework relies on the use of VO (cf. Section 1) to generate the true values of the velocities we employ to train our deep network. Visual odometry computes estimates of the robot's pose from a set of camera images by analysing the variations on those images generated by the motion of the vehicle. Different approaches and implementations have been proposed in the literature (for a review, see [12]). In this work, we utilise a stereo VO pipeline which makes use of a combination of FAST [13] corners and BRIEF descriptors [14], and applies RANSAC [15] for outlier rejection. The ego-motion is, then, computed by non-linear least squares optimisation. VO provides 6DoF pose estimates. In this case, as the motion of the robot is actually restricted to the ground plane, its pose is fully described by two translational components and one rotational component, which we use to compute the velocities.

3.2 Architecture

Our regression system is based on a DNN which takes as input a feature representation of the robot's ego-noise (cf. Section 3.3) and returns estimates for both the linear and angular velocity at which the robot is travelling. We leverage multi-task learning (MTL) to simultaneously regress both the linear and the angular velocity. Multi-task learning, indeed, allows greater generalisation, as it is able to take advantage of information in training signals of related tasks, and has been successfully used in several applications [16, 17]. In this work, we opt for *hard parameter sharing*, which was firstly introduced by [18]. In the resulting architecture, the input and the first hidden layer are shared across the two tasks (i.e. linear and angular velocity estimation), while the rest of the hidden layers are not shared. This architecture was empirically chosen, as being the one yielding the best performance on our dataset. We employ for both tasks four hidden layers with a Rectified Linear (ReL) function. Our network outputs a vector $\mathbf{V} = [\hat{v}, \hat{\omega}]$, consisting of the estimates of the magnitude of the linear velocity $\hat{v}$ and the magnitude of the angular velocity $\hat{\omega}$. Similarly to [5], we define our loss function L as:

$$L = \|v - \hat{v}\|_2 + \|\omega - \hat{\omega}\|_2 \tag{1}$$

where v and ω are the ground truth values we can extrapolate from our Visual Odometry system (cf. Section 3.1) . Training is performed by minimising the Euclidean loss L with $l1$ regularisation, using back-propagation.

3.3 Feature Representation

In audio-based classification, Mel-frequency cepstrum coefficients (MFCCs) [6] have been traditionally used as feature representations of the signals. However, recent studies proved that the performance of classification systems relying on MFCCs is greatly reduced in the presence of noise [7, 19]. Our data is affected by some environmental noise, such as people talking in the proximity of the robot, construction works nearby, wind, and cars passing. Noise which is especially manifest when the vehicle is moving slowly, as the sound level (i.e. volume) of the motion tends to increase with the speed. In the attempt of being more resilient to these additional and unwanted acoustic signals, in this work we opt for a frequency representation based on Gammatone filterbanks, which have been originally introduced in [20], as an approximation to the human cochlear frequency selectivity, and later used in several contexts (e.g. [21]). Similarly to [7], we employ a time-independent representation of the sound, which is obtained by filtering the audio waveform with a bank of Gammatone band-pass filters. The impulse response of a Gammatone filter centered at frequency f_c is:

$$g(t, f_c) = \begin{cases} t^{a-1} e^{-2\pi b t} \cos 2\pi f_c t & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where a indicates the order of the filter, and b is the bandwidth, which increases as the center frequency f_c increases. The frequency-dependent bandwidth yield narrower filters at low frequencies and broader filters at high frequencies. Several investigations have been carried out to compute the values of the filters' parameters which best approximate the human auditory filter. In this work, following [22], we utilise fourth-order filters (i.e. $a = 4$), and approximate b as:

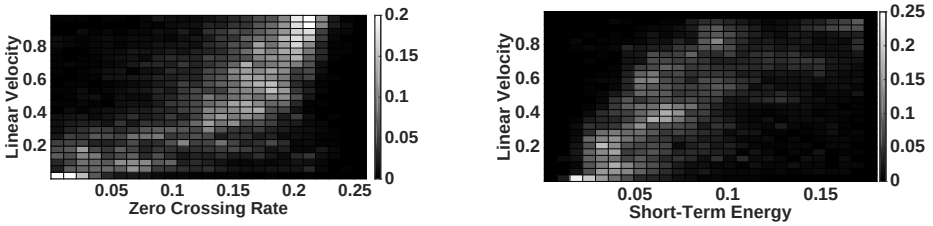
$$b = 1.09 \left(\frac{f_c}{9.26449} + 24.7 \right) \tag{3}$$

The center frequencies f_c of the filters are distributed across the available spectrum in proportion to their bandwidth. The identification of those frequencies can be achieved by using the Equivalent Rectangular Bandwidth (ERB) scale [23]. Let $x(t)$ be the audio signal we want to analyse, the output response $y(t, f_c)$ of a filter characterised by the center frequency f_c can be computed as:

$$y(t, f_c) = x(t) * g(t, f_c) \tag{4}$$

We calculate the output response for all the filters in the bank. The energy of these output responses, expressed in dB, represents our feature representation of the audio signal in the frequency domain, which we name GTF. Extending the same procedure to overlapping time frames of the signal, it is possible to generate time-frequency representations which follow the frequency resolution imposed by the Gammatone filterbank, the *Gammatonegrams*. Examples of the gammatonegrams and GTF representations for frames of 1 s, recorded with the robot travelling at different linear velocities (angular velocity is negligible in those frames) are provided in Figure 2 and 3.
Additionally, we consider also some signal statistics in the time domain, such as the

short-term energy (STE) of an entire frame and the zero-crossing rate (ZCR). The zero-crossing rate indicates the number of times the signal changes its sign within a frame. Figure 1 shows a 2D normalised histogram of the ZCR and linear velocity pairs (Figure 1a), as well as a 2D normalised histogram of the short-term energy and linear velocity pairs (Figure 1b). We notice that the ZCR increases with the linear velocity. The same is observed for what concerns the short-term energy. Both considerations suggest that higher linear velocity are characterised by higher frequencies (due to the higher ZCR) and by a higher sound level (due to higher STE). Same behaviour is appreciable from the Gammatone filterbanks and the GTF representations (cf. Figures 2 and 3). While no apparent pattern is observed for what concerns the angular velocity and either the ZRC or the STE, an example of how the angular velocity affects the spectrum of the ego-noise is presented in Figure 4. The figure shows the time-independent representation of two frames, characterised by the same linear velocity ($v = 0.4$ m/s) and different angular one ($\omega \in \{0.005, 0.26\}$ rad/s). We notice that the angular velocity mainly influences the lower part of the ego-noise's frequency spectrum. In particular, a higher angular velocity is reflected into greater energy in the lower part of the frequency spectrum. The complete framework is shown in Figure 5.



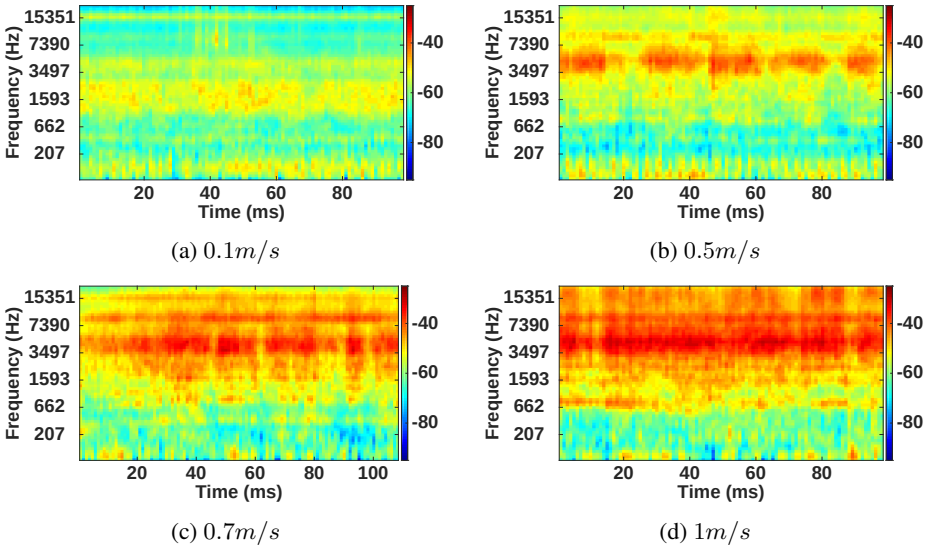
(a) ZCR vs linear velocity

(b) STE vs linear velocity

Fig. 1: From left to right: 2D normalised histogram of the zero crossing rate (ZCR) and linear velocity pairs (a) and 2D normalised histogram of the short-term energy (STE) and linear velocity pairs (b). We notice that the ZCR increases with the linear velocity. The same is observed for what concerns the STE.

4 Experimental Evaluation

To validate our framework, we collected approximately two hours of data at the University Parks in Oxford, UK, using a Clearpath Husky A200 platform. The robot is equipped with a Bumblebee2 stereo camera, two Knowles omnidirectional boom microphones mounted in proximity of each of the two front wheels, and an ALESIS IO4 audio interface. The stereo camera is used to collect the image data that will be employed by the VO pipeline to generate ground truth values for the velocity estimates. The audio data has been recorded at a sampling frequency f_s of 44100 Hz at a resolution of 16 bits. Camera images are gathered at a rate of 10 frames per second (FPS) and with 768×1024 pixel resolution. The data covers a total route of about 2 Km in length and includes portions of the park characterised by different kinds of terrain, such as grass, soil, and gravel. We made this choice to build a regression model able to generalise to different surfaces. The data was collected by manually driving the platform



(a) $0.1m/s$ (b) $0.5m/s$

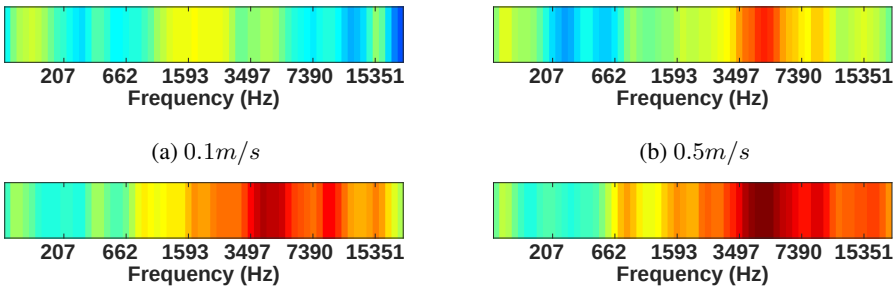
(c) $0.7m/s$ (d) $1m/s$

Fig. 2: Gammatonegrams of $1s$ frames of ego-noise recorded with the robot travelling at different linear velocities: (a) robot is travelling at 0.1 m/s, (b) robot travelling at 0.5 m/s, (c) robot travelling at 0.7 m/s, (d) robot travelling at 1 m/s. Angular velocity is negligible in those frames. The energy of the time-frequency bins is expressed in decibel (dB) scale. The filtering is performed using 64 frequency channels (i.e. number of filters in the Gammatone filterbank) between 0 Hz and 22050 Hz.

to obtain a wider spectrum of motion profiles and be able to encapsulate the behaviour of the robot in several circumstances. We used in total $85K$ frames of $1s$ for training and $13K$ for testing. We carry out two different kinds of experiments. We first evaluate our MTL framework, varying the number of frequency channels used in the frequency representation. In particular, we consider 64 and 128 frequency channels (i.e. number of filters in the Gammatone filterbank). Secondly, we compare the behaviour of the MTL framework with a baseline one, represented by a single-task learning scheme, where the two velocities are regressed separately by two different networks.

4.1 Implementation Details

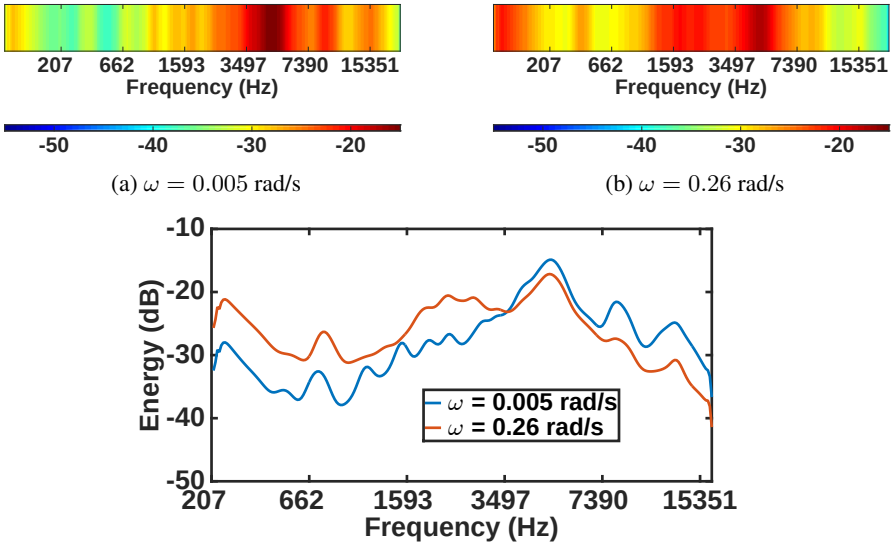
We trained the networks using mini-batch gradient descent based on back propagation, employing the Adam optimisation algorithm [24]. We applied dropout [25] to each non-shared layer for both tasks' architectures with a keeping probability of 0.9. The models were implemented using the Tensorflow [26] libraries. Independently of the number of filters utilised, we confine our frequency analysis to a range between 0 Hz and $f_s/2 = 22050$ Hz, corresponding to the maximum reliable frequency resolution available. The filtering is computed on time domain frames of 1 s with 10 ms overlap, after applying a Hamming window to avoid spectral leakage. As our VO system returns ego-motion estimates at 10 Hz, we can generate the 1 s audio frames by using a sliding



(a) $0.1m/s$ (b) $0.5m/s$

(c) $0.7m/s$ (d) $1m/s$

Fig. 3: GTF representations correspondent to the gammatonegrams in Figure 2: (a) robot is travelling at 0.1 m/s, (b) robot travelling at 0.5 m/s, (c) robot travelling at 0.7 m/s, (d) robot travelling at 1 m/s. Angular velocity is negligible in those frames. The filtering is performed using 64 frequency channels (i.e. number of filters in the Gammatone filterbank) between 0 Hz and 22050 Hz.



(a) $\omega = 0.005$ rad/s (b) $\omega = 0.26$ rad/s

(c) Energy vs Frequency Channels

Fig. 4: The figure shows the GTF representations of two frames, characterised by the same linear velocity ($v = 0.4$ m/s) and different angular ones. Specifically, the top-left diagram (a) represents an audio frame where the vehicle is travelling at $\omega = 0.005$ rad/s, while the top-right (b) represents an audio frame where the vehicle is travelling at $\omega = 0.26$ rad/s. Energy is expressed in dB. Differences between the frequency spectra of the two frames are further highlighted in (c). We observe that a higher angular velocity is reflected into greater energy in the lower part of the frequency spectrum.

window of 1 s size with 100 ms overlap. Similarly to previous works on deep learning in the auditory domain (cf. [27], [28]), we randomly split our dataset into training set (85%) and test set (15%). To avoid any unwanted overlap between training and testing sets, frames in the training set and frames in the test set do not share any audio segment across the sliding windows.

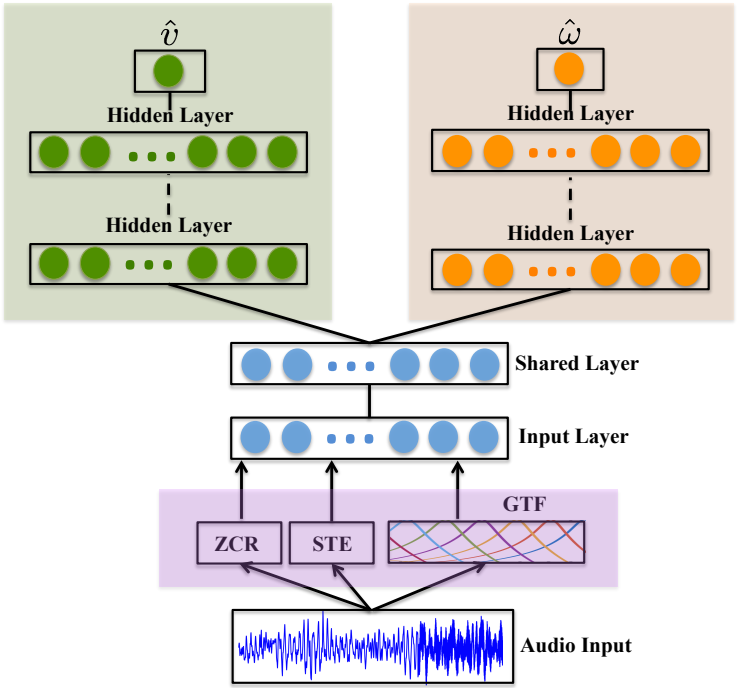
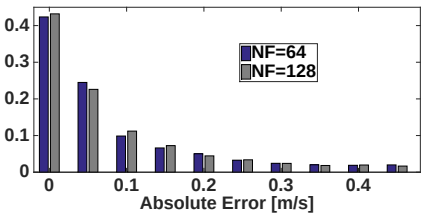


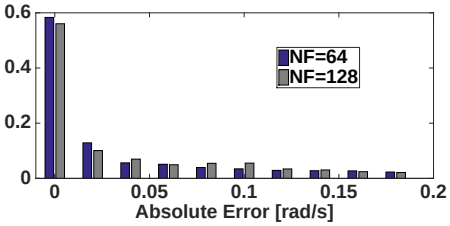
Fig. 5: Representation of the learning scheme used. The waveform of the original signal in the time domain is parsed and features are extracted (purple area). Specifically, ZCR indicates the zero-crossing rate of the signal, STE represents the short-term energy, and GTF is the feature representation in the frequency domain obtained after applying the Gammatone filterbank to the original signal. Features are then concatenated and fed to the DNN. The input and the first hidden layer of the networks are shared across the two task (i.e. linear and angular velocity estimation). The rest of the hidden layers (we used four in this case for both tasks) are not shared. The green structure refers to the portion of the network employed to regress exclusively the linear velocity $\hat{v}$, while the orange one refers to the portion of the network employed to regress the angular velocity $\hat{\omega}$. All hidden layers are equipped with a Rectified Linear Unit (ReLU).

4.2 Experiment 1: Multi-Task Learning

In this first experiment we analyse the performance of our system whilst varying the number of filters NF employed. In particular, we evaluate the framework for $NF \in \{64, 128\}$. Table 1 reports the results of this experiment. In the table, $\tilde{E_v}$ and $\tilde{E_\omega}$ refer to the median absolute error in the estimation of the linear and angular velocity on the test data. No significant difference in the performance is observable when increasing the number of filters from 64 to 128, neither for what concerns the linear velocity nor in case of the angular velocity, leading to the conclusion that 64 filters have already a proper representation power for the task. The table also reports the median absolute error in the estimation of the linear and angular velocity in the MTL regression framework on the training data, indicated with $\tilde{T_v}$ and $\tilde{T_\omega}$, respectively. Figure 6 shows the normalised histograms of the absolute error in the estimation of the linear velocity (Figure 6a) and in the estimation of the angular velocity (Figure 6b), for $NF \in \{64, 128\}$, when following an MTL scheme.



(a) Normalised Histogram of E_v in the MTL case



(b) Normalised Histogram of E_ω in the MTL case

Fig. 6: From left to right: Normalised histogram of the absolute error in the estimation of the linear velocity E_v in the MTL case (a), Normalised histogram of the absolute error in the estimation of the angular velocity E_ω in the MTL case (b). Both histograms refer to the errors on the testing data. NF refers to the number of frequency channels used in the feature representation.

4.3 Experiment 2: Single-Task Learning

In this second experiment we compare the behaviour of the MTL framework with a baseline one, represented by a single-task learning scheme. Specifically, in the STL case, we employ the same deep architecture as the one illustrated in Figure 5, but without the shared layer, i.e. we regress the two velocities separately. Also in this experiment, we consider two different feature representations, obtained by using either 64 or 128 frequency channels. Table 1 reports the results of this experiment. In the table, $\tilde{E_{v_s}}$ and $\tilde{E_{\omega_s}}$ refer to the median absolute error in the estimation of the linear and angular velocity on the test data, by using the STL regression framework. $\tilde{T_{v_s}}$ and $\tilde{T_{\omega_s}}$ indicate the median absolute error in the estimation of the linear and angular velocity on the training data. Figure 7 shows the normalised histograms of the absolute error in the estimation of the linear velocity (Figure 7a) and in the estimation of the angular velocity

(Figure 7b), for $NF \in \{64, 128\}$, when following an STL scheme. We see that in this case, employing 64 frequency channels yields better performance in the linear velocity estimation, while the opposite behaviour is reported for what concerns the angular velocity.

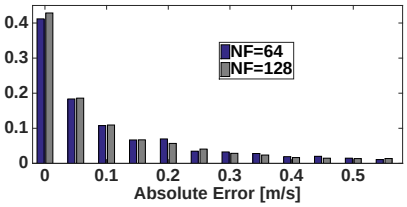
NF	**MTL**				**STL**			
	$\tilde{E_v}$ [m/s]	$\tilde{E_\omega}$ [rad/s]	$\tilde{T_v}$ [m/s]	$\tilde{T_\omega}$ [rad/s]	$\tilde{E_v}$ [m/s]	$\tilde{E_\omega}$ [rad/s]	$\tilde{T_v}$ [m/s]	$\tilde{T_\omega}$ [rad/s]
64	**0.065**	**0.017**	0.041	0.013	0.074	0.023	0.037	0.011
128	**0.064**	**0.018**	0.040	0.009	0.081	0.021	0.030	0.009

Table 1: The table reports the results of the experiments. $\tilde{E_v}$ and $\tilde{E_\omega}$ indicate the median absolute error in the estimation of the linear and angular velocity in the MTL and the STL regression frameworks on the test data. $\tilde{T_v}$ and $\tilde{T_\omega}$ indicate the median absolute error in the estimation of the linear and angular velocity in the MTL and the STL regression frameworks on the training data. NF refers to the number of filters used in the Gammatone filterbank.

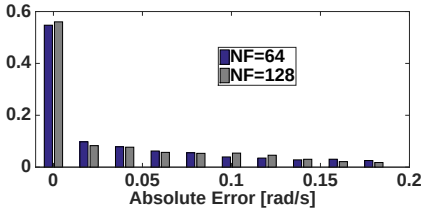
When comparing the behaviour of the STL system with the MTL one, we observe that the MTL scheme outperforms the STL one, independently of the number of channels used, both in the case of the linear and the angular velocity. Specifically, we obtain an improvement in the performance by 12% on $\tilde{E_v}$ and 22% on $\tilde{E_\omega}$, when $NF = 64$, and an improvement by 26% on $\tilde{E_v}$ and 14% on $\tilde{E_\omega}$, when $NF = 128$. We also notice that the STL scheme is characterised by a lower error on the training set, in the estimation of both velocities, independently of the number of filters used. Such a behaviour is expected, as one of the advantages of MTL, especially in case of hard parameter sharing, is, indeed, to help against overfitting, increasing the generalisation capabilities of the model.

5 Conclusions

In this paper we investigated the possibility of estimating a robot's ego-motion by relying only on acoustic sensing. We performed regression analysis employing a deep neural network and followed a multi-task learning scheme to simultaneously estimate the magnitude of the linear and the angular velocity of the vehicle. Our experimental evaluation conducted on approximately two hours of data collected by an unmanned outdoor field robot proved that our framework is able to provide accurate ego-motion estimates, despite the presence of background noise and the robot travelling on different kinds of terrain. When compared to a single-task learning scheme, where the two velocities are modelled separately, our framework shows an improvement of up to 26% in the ego-motion estimation. Given those results, we envision this system being useful



(a) Normalised Histogram of E_v in the STL case



(b) Normalised Histogram of E_ω in the STL case

Fig. 7: From left to right: Normalised histogram of the absolute error in the estimation of the linear velocity E_v in the STL case (a), Normalised histogram of the absolute error in the estimation of the angular velocity E_ω in the STL case (b). Both histograms refer to the errors on the testing data. NF refers to the number of frequency channels used in the feature representation.

as an auxiliary source of odometry information on the side of more traditional odometry systems. Acoustic sensing, indeed, is not affected by lighting or scene appearance. Future work could investigate the possibility of combining the current framework with visual odometry or laser-based odometry systems in a multi-modal setting.

Acknowledgements: This work was supported by the UK EPSRC Programme Grant EP/M019918/1.

References

1. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry for ground vehicle applications. Journal of Field Robotics **23**(1) (2006) 3–20
2. Maimone, M., Cheng, Y., Matthies, L.: Two years of visual odometry on the mars exploration rovers. Journal of Field Robotics **24**(3) (2007) 169–186
3. Nicolai, A., Skeele, R., Eriksen, C., Hollinger, G.A.: Deep learning for laser based odometry estimation. Robotics: Science and Systems, Workshop on Limits and Potentials of Deep Learning in Robotics (2016)
4. Konda, K.R., Memisevic, R.: Learning visual odometry with a convolutional network. In: VISAPP (1). (2015) 486–490
5. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE international conference on computer vision. (2015) 2938–2946
6. Marchegiani, L., Pirri, F., Pizzoli, M.: Multimodal speaker recognition in a conversation scenario. In: International Conference on Computer Vision Systems, Springer (2009) 11–20
7. Marchegiani, L., Posner, I.: Leveraging the urban soundscape: Auditory perception for smart vehicles. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on, IEEE (2017) 6547–6554
8. Maxime, J., Alameda-Pineda, X., Girin, L., Horaud, R.: Sound representation and classification benchmark for domestic robots. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE (2014) 6285–6292
9. Valada, A., Spinello, L., Burgard, W.: Deep feature learning for acoustics-based terrain classification. In: Robotics Research. Springer (2018) 21–37

10. Bando, Y., Suhara, H., Tanaka, M., Kamegawa, T., Itoyama, K., Yoshii, K., Matsuno, F., Okuno, H.G.: Sound-based online localization for an in-pipe snake robot. In: Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on, IEEE (2016) 207–213
11. Pico, A., Schillaci, G., Hafner, V.V., Lara, B.: How do i sound like? forward models for robot ego-noise prediction. In: Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2016 Joint IEEE International Conference on, IEEE (2016) 246–251
12. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. IEEE robotics & automation magazine **18**(4) (2011) 80–92
13. Rosten, E., Reitmayr, G., Drummond, T.: Real-time video annotations for augmented reality. Advances in Visual Computing (2005) 294–302
14. Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., Fua, P.: Brief: Computing a local binary descriptor very fast. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(7) (2012) 1281–1298
15. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6) (1981) 381–395
16. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning, ACM (2008) 160–167
17. Jin, F., Sun, S.: Neural network multitask learning for traffic flow forecasting. In: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, IEEE (2008) 1897–1901
18. Caruana, R.: Multitask learning. In: Learning to learn. Springer (1998) 95–133
19. Chakrabarty, D., Elhilali, M.: Abnormal sound event detection using temporal trajectories mixtures. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE (2016) 216–220
20. Holdsworth, J., Nimmo-Smith, I., Patterson, R., Rice, P.: Implementing a gammatone filter bank. Annex C of the SVOS Final Report: Part A: The Auditory Filterbank **1** (1988) 1–5
21. Marchegiani, L., Karadogan, S.G., Andersen, T., Larsen, J., Hansen, L.K.: The role of top-down attention in the cocktail party: Revisiting cherry's experiment after sixty years. In: Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on. Volume 1., IEEE (2011) 183–188
22. Toshio, I.: An optimal auditory filter. In: Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on, IEEE (1995) 198–201
23. Glasberg, B.R., Moore, B.C.: Derivation of auditory filter shapes from notched-noise data. Hearing research **47**(1) (1990) 103–138
24. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) Published as a conference paper at the 3rd International Conference for Learning Representations (ICLR) 2015.
25. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
26. et al, M.A.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015) Software available from tensorflow.org.
27. Deng, S., Han, J., Zhang, C., Zheng, T., Zheng, G.: Robust minimum statistics project coefficients feature for acoustic environment recognition. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE (2014) 8232–8236
28. Takahashi, N., Gygli, M., Van Gool, L.: Aenet: Learning deep audio features for video analysis. arXiv preprint arXiv:1701.00599 (2017)