Scale-Adaptive Neural Dense Features: Learning via Hierarchical Context Aggregation

Jaime Spencer, Richard Bowden, Simon Hadfield
Centre for Vision, Speech and Signal Processing (CVSSP)
University of Surrey
{jaime.spencer, r.bowden, s.hadfield}@surrey.ac.uk

Abstract

How do computers and intelligent agents view the world around them? Feature extraction and representation constitutes one the basic building blocks towards answering this question. Traditionally, this has been done with carefully engineered hand-crafted techniques such as HOG, SIFT or ORB. However, there is no "one size fits all" approach that satisfies all requirements.

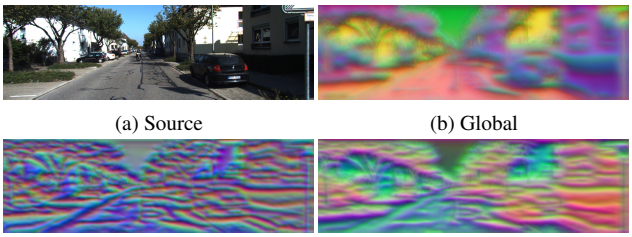
In recent years, the rising popularity of deep learning has resulted in a myriad of end-to-end solutions to many computer vision problems. These approaches, while successful, tend to lack scalability and can't easily exploit information learned by other systems.

Instead, we propose SAND features, a dedicated deep learning solution to feature extraction capable of providing hierarchical context information. This is achieved by employing sparse relative labels indicating relationships of similarity/dissimilarity between image locations. The nature of these labels results in an almost infinite set of dissimilar examples to choose from. We demonstrate how the selection of negative examples during training can be used to modify the feature space and vary it's properties.

To demonstrate the generality of this approach, we apply the proposed features to a multitude of tasks, each requiring different properties. This includes disparity estimation, semantic segmentation, self-localisation and SLAM. In all cases, we show how incorporating SAND features results in better or comparable results to the baseline, whilst requiring little to no additional training. Code can be found at: https://github.com/jspenmar/SAND_features

1. Introduction

Feature extraction and representation is a fundamental component of most computer vision research. We propose to learn a feature representation capable of supporting a wide range of computer vision tasks. Designing such a system proves challenging, as it requires these features to be both unique and capable of generalizing over radical changes in appearances at the pixel-level. Areas such as Simultaneous Localisation and Mapping (SLAM) or Visual Odometry (VO) tend to use feature extraction in an explicit manner [2, 17, 20, 46], where hand-crafted sparse features are extracted from pairs of images and matched against each other. This requires globally consistent and unique features that are recognisable from wide baselines.



(a) Source (b) Global (c) Local (d) Hierarchical

Figure 1: Visualization of SAND features trained using varying context hierarchies to target specific properties.

On the other hand, methods for optical flow [32] or object tracking [3] might instead favour locally unique or smooth feature spaces since they tend to require iterative processes over narrow baselines. Finally, approaches typically associated with deep learning assume feature extraction to be implicitly included within the learning pipeline. End-to-end methods for semantic segmentation [6], disparity estimation [44] or camera pose regression [29] focus on the learning of implicit "features" specific to each task.

Contrary to these approaches, we treat feature extraction as it's own separate deep learning problem. By employing sparsely labelled correspondences between pairs of images, we explore approaches to automatically learn dense representations which solve the correspondence problem while exhibiting a range of potential properties. In order to learn from this training data, we extend the concept of contrastive loss [16] to pixel-wise non-aligned data. This results in a fixed set of positive matches from the ground truth correspondences between the images, but leaves an almost infinite range of potential negative samples. We show how by carefully targeting specific negatives, the properties of the

learned feature representations can be modified to adapt to multiple domains, as shown in Figure 1. Furthermore, these features can be used in combination with each other to cover a wider range of scenarios. We refer to this framework as Scale-Adaptive Neural Dense (SAND) features.

Throughout the remainder of this paper we demonstrate the generality of the learned features across several types of computer vision tasks, including stereo disparity estimation, semantic segmentation, self-localisation and SLAM. Disparity estimation and semantic segmentation first combine stereo feature representations to create a 4D cost volume covering all possible disparity levels. The resulting cost volume is processed in a 3D stacked hourglass network [5], using intermediate supervision and a final upsampling and regression stage. Self-localisation uses the popular PoseNet [19], replacing the raw input images with our dense 3D feature representation. Finally, the features are used in a sparse feature matching scenario by replacing ORB/BRIEF features in SLAM [33].

Our contributions can be summarized as follows:

1. We present a methodology for generic feature learning from sparse image correspondences.

2. Building on "pixel-wise" contrastive losses, we demonstrate how targeted negative mining can be used to alter the properties of the learned descriptors and combined into a context hierarchy.

3. We explore the uses for the proposed framework in several applications, namely stereo disparity, semantic segmentation, self-localisation and SLAM. This leads to better or comparable results in the corresponding baseline with reduced training data and little or no feature finetuning.

2. Related Work

Traditional approaches to matching with hand-engineered features typically rely on sparse keypoint detection and extraction. SIFT [24] and ORB [35], for example, still remain a popular and effective option in many research areas. Such is the case with ORB-SLAM [27, 28] and its variants for VO [12, 48] or visual object tracking methods, including Sakai *et al*. [36], Danelljan *et al*. [25, 8] or Wu *et al*. [43]. In these cases, only globally discriminative features are required, since the keypoint detector can remove local ambiguity.

As an intermediate step to dense feature learning, some approaches aim to learn both keypoint detection and feature representation. Most methods employ hand-crafted feature detectors as a baseline from which to collect data, such as [1, 21]. Alternative methods include Salti *et al*. [37], who treat keypoint detection as a binary classification task, and Georgakis *et al*. [15], who instead propose a joint end-to-end detection and extraction network.

On the other hand, most approaches to dedicated feature learning tend to focus on solving dense correspondence estimation rather than using sparse keypoints. Early work in this area did not perform explicit feature extraction and instead learns a task specific latent space. Such is the case with end-to-end VO methods [42, 22], camera pose regression [19, 4] or stereo disparity estimation [47]. Meanwhile, semantic and instance segmentation approaches such as those proposed by Long *et al*. [23], Noh *et al*. [31] or Wang *et al*. [41] produce a dense representation of the image containing each pixel's class. These require dense absolute labels describing specific properties of each pixel. Despite advances in the annotation tools [9], manual checking and refinement still constitutes a significant burden.

Relative labels, which describe the relationships of similarity or dissimilarity between pixels, are much easier to obtain and are available in larger quantities. Chopra *et al*. [7], Sun *et al*. [40] and Kang *et al*. [18] apply these to face re-identification, which requires learning a discriminative feature space that can generalize over a large amount of unseen data. As such, these approaches make use of relational learning losses such as contrastive [16] or triplet loss [39]. Further work by Yu *et al*. [45] and Ge *et al*. [13] discusses the issues caused by triplet selection bias and provides methods to overcome them.

As originally presented, these losses don't tackle dense image representation and instead compare holistic image descriptors. Schmidt *et al*. [38] propose a "pixel-wise" contrastive loss based on correspondences obtained from KinectFusion [34] and DynamicFusion [30]. Fathy *et al*. [10] incorporate an additional matching loss for intermediate layer representations. More recently, the contextual loss [26] has been proposed as a similarity measure for non-aligned feature representations. In this paper we generalise the concept of "pixel-wise" contrastive loss to generic correspondence data and demonstrate how the properties of the learned feature space can be manipulated.

3. SAND Feature Extraction

The aim of this work is to provide a high-dimensional feature descriptor for every pixel within an image, capable of describing the context at multiple scales. We achieve this by employing a pixel-wise contrastive loss in a siamese network architecture.

Each branch of the siamese network consists of a series of convolutional residual blocks followed by a Spatial Pooling Pyramid (SPP) module, shown in Figure 2. The convolution block and base residual blocks serve as the initial feature learning. In order to increase the receptive field, the final two residual blocks employ an atrous convolution with dilations of two and four, respectively.

The SPP module is formed by four parallel branches, each with average pooling scales of 8, 16, 32 and 64, respectively. Each branch produces a 32D output with a resolution

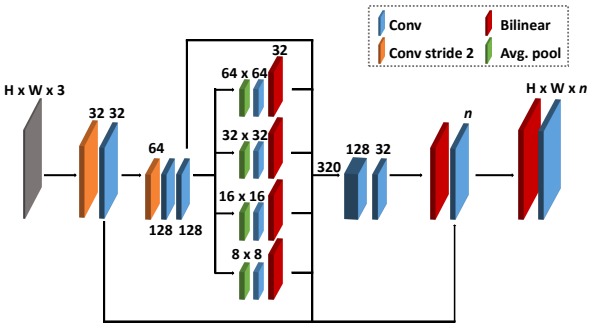


Figure 2: SAND architecture trained for dense feature extraction. The initial convolutions are residual blocks, followed by a 4-branch SPP module and multi-stage decoder.

of $(H/4, W/4)$. In order to produce the final dense feature map, the resulting block is upsampled in several stages incorporating skip connections and reducing it to the desired number of dimensions, n.

Given an input image I, it's dense n-dimensional feature representation can be obtained by

$$F(\boldsymbol{p}) = \Phi(I(\boldsymbol{p})|w), \tag{1}$$

where $\boldsymbol{p}$ represents a 2D point and Φ represents a SAND branch, parametrized by a set of weights w. I stores RGB colour values, whereas F stores n-dimensional feature descriptors, $\Phi : \mathbb{N}^3 \rightarrow \mathbb{R}^n$.

3.1. Pixel-wise Contrastive Loss

To train this feature embedding network we build on the ideas presented in [38] and propose a pixel-wise contrastive loss. A siamese network with two identical SAND branches is trained using this loss to produce dense descriptor maps. Given a pair of input points, contrastive loss is defined as

$$l(y, \boldsymbol{p}_1, \boldsymbol{p}_2) = \begin{cases} \frac{1}{2}(d)^2 & \text{if } y = 1 \\ \frac{1}{2}\{\max(0, m-d)\}^2 & \text{if } y = 0 \\ 0 & otherwise \end{cases} \tag{2}$$

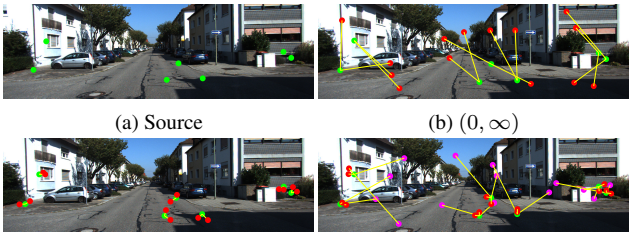
where d is the euclidean distance of the feature embeddings $||\boldsymbol{F}_1(\boldsymbol{p}_1) - \boldsymbol{F}_2(\boldsymbol{p}_2)||$, y is the label indicating if the pair is a match and m is the margin. Intuitively, positive pairs (matching points) should be close in the latent space, while negative pairs (non-matching points) should be separated by at least the margin.

The labels indicating the similarity or dissimilarity can be obtained through a multitude of sources. In the simplest case, the correspondences are given directly by disparity or optical flow maps. If the data is instead given as homogeneous 3D world points $\dot{\boldsymbol{q}}$ in a depth map or pointcloud, these can be projected onto pairs of images. A set of corresponding pixels can be obtained through

$$\boldsymbol{p} = \pi(\dot{\boldsymbol{q}}) = \boldsymbol{K}\boldsymbol{P}\dot{\boldsymbol{q}}, \tag{3}$$

$$(\boldsymbol{c}_1, \boldsymbol{c}_2) = (\boldsymbol{p}_1, \boldsymbol{p}_2) \text{ where } \pi_1(\dot{\boldsymbol{q}}) \mapsto \pi_2(\dot{\boldsymbol{q}}), \tag{4}$$

where π is the projection function parametrized by the corresponding camera's intrinsics $\boldsymbol{K}$ and global pose $\boldsymbol{P}$.



(a) Source (b) $(0, \infty)$

(c) $(0, 25)$ (d) $(0, \infty)$ - $(0, 25)$

Figure 3: Effect of (α, β) thresholds on the scale information observed by each individual pixel. Large values of α and β favour global features, while low β values increase local discrimination.

A label mask $\boldsymbol{Y}$ is created indicating if every possible combination of pixels is a positive example, negative example or should be ignored. Unlike a traditional siamese network, every input image has many matches, which are not spatially aligned. As an extension to (2) we obtain

$$L(\boldsymbol{Y}, \boldsymbol{F}_1, \boldsymbol{F}_2) = \sum_{\boldsymbol{p}_1} \sum_{\boldsymbol{p}_2} l(\boldsymbol{Y}(\boldsymbol{p}_1, \boldsymbol{p}_2), \boldsymbol{p}_1, \boldsymbol{p}_2). \tag{5}$$

3.2. Targeted Negative Mining

The label map $\boldsymbol{Y}$ provides the list of similar and dissimilar pairs used during training. The list of similar pairs is limited by the ground truth correspondences between the input images. However, each of these points has $(H \times W) - 1$ potential dissimilar pairs $\hat{\boldsymbol{c}}_2$ to choose from. This only increases if we consider all potential dissimilar pairs within a training batch. For converting 3D ground truth data we can define an equivalent to (4) for negative matches,

$$\hat{\boldsymbol{c}}_2 \sim \boldsymbol{p}_2 \text{ where } \pi_1^{-1}(\boldsymbol{c}_1) \nleftrightarrow \pi_2^{-1}(\boldsymbol{p}_2). \tag{6}$$

It is immediately obvious that it is infeasible to use all available combinations due to computational cost and balancing. In the naïve case, one can simply select a fixed number of random negative pairs for each point with a ground truth correspondence. By selecting a larger number of negative samples, we can better utilise the variability in the available data. It is also apparent that the resulting highly unbalanced label distributions calls for loss balancing, where the losses attributed to negative samples are inversely weighted according to the total number of pairs selected.

In practice, uniform random sampling serves to provide globally consistent features. However, these properties are not ideal for many applications. By instead intelligently targeting the selection of negative samples we can control the properties of the learned features.

Typically, negative mining consists of selecting hard examples, *i.e.* examples that produce false positives in the network. Whilst this concept could still be applied within the proposed method, we instead focus on spatial mining strategies, as demonstrated in Figure 3. The proposed mining strategy can be defined as

$$\hat{\boldsymbol{c}'}_2 \sim \hat{\boldsymbol{c}}_2 \text{ where } \alpha < ||\hat{\boldsymbol{c}}_2 - \boldsymbol{c}_2|| < \beta. \quad (7)$$

In other words, the negative samples are drawn from a region within a radius with lower and higher bounds of (α, β), respectively. As such, this region represents the area in which the features are required to be unique, *i.e*. the *scale* of the features.

For example, narrow baseline stereo requires locally discriminative features. It is not important for distant regions to be distinct as long as fine details cause measurable changes in the feature embedding. To encourage this, only samples within a designated radius, *i.e*. a small β threshold, should be used as negative pairs. On the other hand, global descriptors can be obtained by ignoring nearby samples and selecting negatives exclusively from distant image regions, *i.e*. large α and $\beta = \infty$.

3.3. Hierarchical Context Aggregation

It is also possible to benefit from the properties of multiple negative mining strategies simultaneously by "splitting" the output feature map and providing each section with different negative sampling strategies. For N_S number of mining strategies, N_C represents the number of channels per strategy, $\lfloor n/N_S \rfloor$. As a modification to (2), we define the final pixel-level loss as

$$l(y, \boldsymbol{p}_1, \boldsymbol{p}_2^1 ... \boldsymbol{p}_2^{N_S}) = \begin{cases} \frac{1}{2}\sum_{i=1}^{N_S} d^2(i) & \text{if } y = 1 \\ \frac{1}{2}\sum_{i=1}^{N_S} \{\max(0, m_i\text{-}d(i))\}^2 & \text{if } y = 0 \\ 0 & otherwise \end{cases} \quad (8)$$

where $\boldsymbol{p}_2^i$ represents a negative sample from strategy i and

$$d^2(i) = \sum_{z=iN_C}^{(i+1)N_C} \left(\boldsymbol{F}_1(\boldsymbol{p}_1, z) - \boldsymbol{F}_2(\boldsymbol{p}_2^i, z)\right)^2. \quad (9)$$

This represents a powerful and generic tool that allows us to further adapt to many tasks. Depending on the problem at hand, we can choose corresponding features scales that best suit the property requirements. Furthermore, more complicated tasks or those requiring multiple types of feature can benefit from the appropriate scale hierarchy. For the purpose of this paper, we will evaluate three main categories: global features, local features and the hierarchical combination of both.

3.4. Feature Training & Evaluation

Training. In order to obtain the pair correspondences required to train the proposed SAND features, we make use of the popular Kitti dataset [14]. Despite evaluating on three of the available Kitti challenges (Odometry, Semantics and Stereo) and the Cambridge Landmarks Dataset, the feature network Φ is pretrained exclusively on a relatively modest subsection of 700 pairs from the odometry sequence 00.

Each of these pairs has 10-15 thousand positive correspondences obtained by projecting 3D data onto the images, with 10 negative samples each, generated using the presented mining approaches. This includes thresholds of $(0, \infty)$ for **G**lobal descriptors, $(0, 25)$ for **L**ocal descriptors and the hierarchical combination of both (**GL**). Each method is trained for 3, 10 and 32 dimensional feature space variants with a target margin of 0.5.

Visualization. To begin, a qualitative evaluation of the learned features can be found in Figure 4. This visualisation makes use of the 3D descriptors, as their values can simply be projected onto the RGB color cube. The exception to this is **GL**, which makes use of 6D descriptors reduced to 3D through PCA. It is immediately apparent how the selected mining process affects the learned feature space.

When considering small image patches, **G** descriptors are found to be smooth and consistent, while they are discriminative regarding distant features. Contrary to this, **L** shows repeated features across the whole image, but sharp contrasts and edges in their local neighbourhood. This aligns with the expected response from each mining method. Finally, **GL** shows a combination of properties from both previous methods.

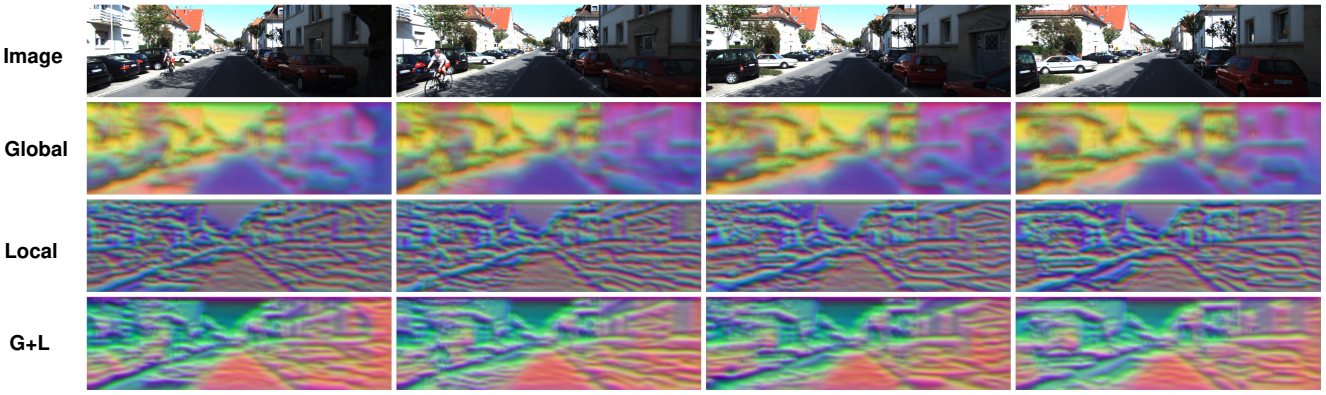


Figure 4: Learned descriptor visualizations for 3D. From top to bottom: source image, **G**lobal mining, 25 pixel **L**ocal mining and hierarchical approach. **L** descriptors show more defined edges and local changes, whereas **GL** provides a combination of both.

D	Mining	μ_+	Global perf.		Local perf.	
			AUC	μ_-	AUC	μ_-
32	**ORB**	NA	85.83	NA	84.06	NA
3	**G**	**0.095**	**98.62**	0.951	84.70	0.300
	L	0.147	96.05	0.628	**91.92**	0.564
	GL (6D)	0.181	97.86	**1.161**	90.67	**0.709**
10	**G**	**0.095**	**99.43**	0.730	86.99	0.286
	L	0.157	98.04	0.579	**93.57**	0.510
	GL	0.187	98.60	**1.062**	91.87	**0.678**
32	**G**	**0.093**	**99.73**	0.746	87.06	0.266
	I	0.120	99.61	0.675	91.94	0.406
	L	0.156	98.88	0.592	**94.34**	0.505
	GL	0.183	99.28	0.996	93.34	0.642
	GIL	0.214	98.88	**1.217**	91.97	**0.784**

Table 1: Feature metrics for varying dimensionality and mining method vs. ORB baseline. **G**lobal and **L**ocal provide the best descriptors in their respective areas, while **GL** and **GIL** maximise the negative distance and provides a balanced matching performance.

Distance distributions. A series of objective measures is provided through the distribution of positive and negative distances in Table 1. This includes a similarity measure for positive examples μ_+ (lower is better) and a dissimilarity measure for negative examples μ_- (higher is better). Additionally, the Area Under the Curve (AUC) measure represents the probability that a randomly chosen negative sample will have a greater distance than the corresponding positive ground truth match. These studies were carried out for both local (25 pixels radius) and global negative selection strategies. Additionally, the 32D features were tested with an intermediate (75 pixel radius) and fully combined **GIL** approach.

From these results, it can be seen that the global approach **G** performs best in terms of positive correspondence representation, since it minimizes μ_+ and maximizes the global AUC across all descriptor sizes. On the other hand, **L** descriptors provide the best matching performance within the local neighbourhood, but the lowest in the global context. Meanwhile, **I** descriptors provide a compromise between **G** and **L**. Similarly, the combined approach provide an intermediate ground where the distance between all negative samples is maximised and the matching performance at all scales is balanced. All proposed variants significantly outperform the shown ORB feature baseline. Finally, it is interesting to note that these properties are preserved across the varying number of dimensions of the learnt feature space, revealing the consistency of the proposed mining strategies.

4. Feature Matching Cost Volumes

Inspired by [5], after performing the initial feature extraction on the stereo images, these are combined in a cost volume ρ by concatenating the left and right features across all possible disparity levels, as defined by

$$\rho(x, y, \delta, z) = \begin{cases} \boldsymbol{F_1}(x, y, z) & \text{if } z \leq n \\ \boldsymbol{F_2}(x + \delta, y, z) & otherwise \end{cases}, \quad (10)$$

where n corresponds to the dimensionality of the feature maps. This results in $\rho(H \times W \times D \times 2n)$, with D representing the levels of disparity. As such, the cost volume provides a mapping from a 4-dimensional index to a single value, $\rho : \mathbb{N}^4 \rightarrow \mathbb{R}$.

It is worth noting that this disparity replicated cost volume represents an application agnostic extension of traditional dense feature matching cost volumes [11]. The following layers are able to produce traditional pixel-wise feature distance maps, but can also perform multi-scale information aggregation and deal with viewpoint variance. The resulting cost volume is fed to a 3D stacked hourglass network composed of three modules. In order to reuse the information learned by previous hourglasses, skip connections are incorporated between corresponding sized layers. As a final modification, additional skip connections from the early feature extraction layers are incorporated before the final upsampling and regression stages.

To illustrate the generality of this system, we exploit the same cost volume and network in two very different tasks. Stereo disparity estimation represents a traditional application for this kind of approach. Meanwhile, semantic segmentation has traditionally made use of a single input image. In order to adapt the network for this purpose, only the final layer is modified to produce an output with the desired number of segmentation classes.

5. Results

In addition to the previously mentioned disparity and semantic segmentation, we demonstrate applicability of the proposed SAND features in two more areas: self-localisation and SLAM. Each of these areas represents a different computer vision problem with a different set of desired properties.

For instance, stereo disparity represents a narrow baseline matching task and as such may favour local descriptors in order to produce sharp response boundaries. Meanwhile, semantic segmentation makes use of implicit feature extraction in end-to-end learned representations. Due to the nature of the problem, feature aggregation and multiple scales should improve performance.

On the other side of the spectrum, self-localisation emphasizes wide baselines and revisitation, where the global appearance of the scene helps determine the likely location. In this case, it is crucial to have globally robust features that are invariant to changes in viewpoint and appearance. Furthermore, the specific method chosen makes use of holistic image representations.

Finally, SLAM has similar requirements to self-localisation, where global consistency and viewpoint invariance is crucial to loop closure and drift minimization. How-

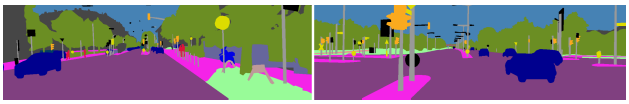
Method	Train (%)	Eval (%)
Baseline [5]	1.49	2.87
10D-G	1.19	3.00
10D-L	1.34	2.82
10D-GL	1.16	2.91
32D-G	**1.05**	**2.65**
32D-L	1.09	2.85
32D-GL	1.06	2.79

Table 2: Disparity error on Kitti Stereo train/eval split. With less training, the proposed methods achieves comparable or better performance than the baseline.

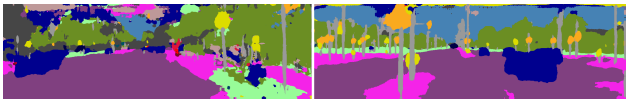
ever, it represents a completely different style of application. In this case, revisitation is detected through sparse direct matching rather than an end-to-end learning approach. Furthermore, the task is particularly demanding of it's features, requiring both wide baseline invariance (mapping) and narrow baseline (VO). As such, it is an ideal use case for the combined feature descriptors.

5.1. Disparity Estimation

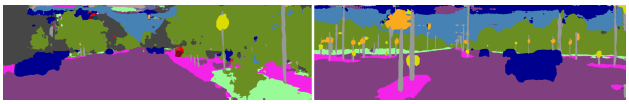
Based on the architecture described in Section 4, we compare our approach with the implementation in [5]. We compare against the original model trained exclusively on the Kitti Stereo 2015 dataset for 600 epochs. Our model fixes the pretrained features for the first 200 epochs and finetunes them at a lower learning rate for 250 epochs. The final error metrics on the original train/eval splits from the public Stereo dataset are found in Table 2 (lower is better).



(a) Ground Truth



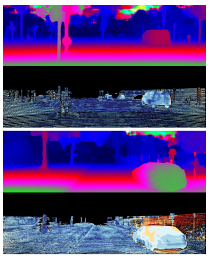
(b) Baseline



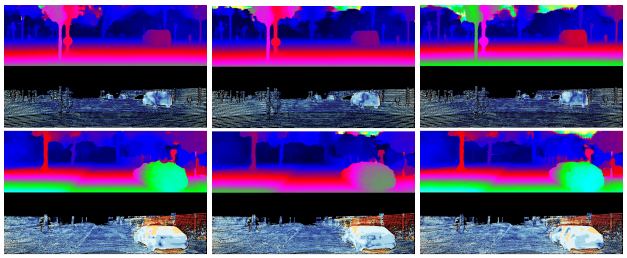
(c) **32D-G-FT**

(d) **32D-GL-FT**

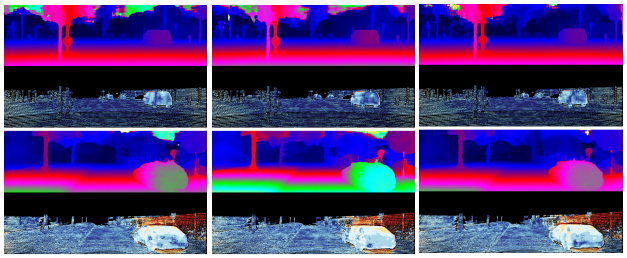
Figure 5: Semantic segmentation visualization for validation set images. The incorporation of SAND features improves the overall level of detail and consistency of the segmented regions.



(a) Baseline



(b) **10D-G** (c) **10D-L** (d) **10D-GL**



(e) **32D-G** (f) **32D-L** (g) **32D-GL**

Figure 6: Disparity visualization for two evaluation images (prediction vs. error). The proposed feature representation increases estimation robustness in complicated areas such as the vehicle windows.

As seen, with 150 less epochs of training the 10D variant achieves a comparable performance, while the 32D variants provide up to a 30% reduction in error. It is interesting to note that **G** features tend to perform better than local and combined approaches, **L** and **GL**. We theorize that the additional skip connections from the early SAND branch make up for any local information required, while the additional global features boost the contextual information. Furthermore, a visual comparison for the results in shown in Figure 6. The second and fourth rows provide a visual representation of the error, where red areas indicate larger errors. As seen in the bottom row, the proposed method increases the robustness in areas such as the transparent car windows.

5.2. Semantic Segmentation

Once again, this approach is based on the cost volume presented in Section 4, with the final layer producing a 19-class segmentation. The presented models are all trained on the Kitti pixel-level semantic segmentation dataset for 600 epochs. In order to obtain the baseline performance,

Method	IoU Class	IoU Cat.	Flat	Nature	Object	Sky	Construction	Human	Vehicle
Baseline	29.3	53.8	87.1	78.1	30.1	**63.3**	54.4	1.6	62.1
32D-G	31.1	55.8	87.3	78.5	36.0	59.8	57.5	**6.7**	66.8
32D-G-FT	**35.4**	**59.9**	**88.7**	83.0	**46.7**	62.7	**63.3**	**6.7**	**68.1**
32D-GL	29.4	51.7	85.1	76.6	33.8	51.8	54.4	4.3	56.3
32D-GL-FT	33.1	56.6	87.4	**91.5**	42.6	56.7	60.4	3.9	63.7

Table 3: Intersection over Union (%) measures for class and category average and per-category breakdown. The incorporation of the proposed features results in an increase in accuracy in complicated categories such as Object and Human.

the stacked hourglass network is trained directly with the input images, whereas the rest use the 32D variants with **G** and **LG** learned features. Unsurprisingly **L** alone does not contain enough contextual information to converge and therefore is not shown in the following results.

In the case of the proposed methods, two SAND variants are trained. The first two fix the features for the first 400 epochs. The remaining two part from these models and finetune the features at a lower learning rate for 200 additional epochs.

As seen in the results in Table 3, the proposed methods significantly outperform the baseline. This is especially the case with Human and Object, the more complicated categories where the baseline fails almost completely. In terms of our features, global features tend to outperform their combined counterpart. Again, this shows that this particular task requires more global information in order to determine what objects are present in the scene than exact location information provided by **L** features.

5.3. Self-localisation

As previously mentioned, self-localisation is performed using the well known method PoseNet [19]. While PoseNet has several disadvantages, including additional training for every new scene, it has proven highly successful and serves as an example application requiring holistic image representation. The baseline was obtained by training a base ResNet34 architecture as described in [19] from scratch with the original dataset images. Once again, the proposed method replaces the input images with their respective SAND feature representation. Both approaches were trained for 100 epochs with a constant learning rate. Once again, only versions denoted **FT** present any additional finetuning to the original pretrained SAND features.

As shown in Table 4, the proposed method with 32D finetuned features generally outperforms the baseline. This contains errors for the regressed position, measured in meters from the ground truth, and rotation representing the orientation of the camera. As expected, increasing the dimensionality of the representation (3 vs. 32) increases the final accuracy, as does finetuning the learnt representations.

Most notably it performs well in sequences like GreatCourt, KingsCollege or ShopFacade. We theorize that this is due to the distinctive features and shapes of the buildings, which allows for a more robust representation. However, the approach tends to perform worse in sequences containing similar or repeating surroundings, such as the Street sequence. This represent a complicated environment for the proposed features in the context of PoseNet, since the global representation can't be reliably correlated with the exact position without additional information.

5.4. SLAM

All previous areas of work explore the use of our features in a deep learning environment, where the dense feature representations are used. This set of experiments instead focuses on their use in a sparse matching domain with explicit feature extraction. The learned features serve as a direct replacement for hand-engineered features. The baseline SLAM system used is an implementation for S-PTAM [33]. This system makes use of ORB descriptors to estimate VO and create the environment maps.

We perform no additional training or adaptation of our features, or any other part of the pipeline for this task. We simply drop our features into the architecture that was built around ORB. It is worth emphasising that we also do not aggregate our features over a local patch. Instead we rely on the feature extraction network to have already encoded all relevant contextual information in the pixel's descriptor.

Method	GreatCourt		KingsCollege		OldHospital		ShopFacade		StMarysChurch		Street	
	P	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*	*P*	*R*
Baseline	10.30	0.35	1.54	0.09	**3.14**	0.10	2.224	**0.19**	**2.77**	**0.22**	**22.60**	1.01
3D-G	12.05	0.33	2.18	0.09	4.07	0.09	2.66	0.29	4.21	0.26	36.13	1.53
32D-G	11.46	0.30	1.62	0.09	3.30	0.11	2.20	0.25	3.67	0.23	31.92	1.24
32D-G-FT	**8.226**	**0.26**	**1.52**	**0.08**	3.21	**0.9**	**2.01**	0.22	3.16	**0.22**	29.89	**0.99**

Table 4: Position (m) and Rotation (deg/m) error for baseline PoseNet vs. SAND feature variants. **FT** indicates a variant with finetuned features. The proposed methods outperforms the baseline in half of the sequence in terms of position error and all except one in terms of rotation error.

Method	00		02		03		04		05	
	APE	RPE	APE	RPE	APE	RPE	APE	RPE	APE	RPE
Baseline	5.63	0.21	**8.99**	**0.28**	6.39	0.05	**0.69**	**0.04**	2.35	0.12
32D-G	13.09	0.21	41.65	0.36	6.00	0.08	6.43	0.13	6.59	0.16
32D-L	5.99	0.21	9.83	0.29	4.40	**0.04**	1.13	0.05	2.37	0.12
32D-GL	**4.84**	**0.20**	9.66	0.29	**3.69**	**0.04**	1.35	0.05	**1.93**	**0.11**
Method	06		07		08		09		10	
Baseline	3.78	0.09	1.10	**0.19**	**4.19**	**0.13**	5.77	0.43	2.06	**0.28**
32D-G	9.10	0.13	2.05	0.21	15.40	0.17	11.50	0.45	18.25	0.35
32D-L	2.54	0.09	**0.88**	**0.19**	5.26	**0.13**	6.25	**0.42**	2.03	0.30
32D-GL	**2.00**	**0.08**	0.96	**0.19**	6.00	**0.13**	**5.48**	**0.42**	**1.36**	0.29

Table 5: Absolute and relative pose error (lower is better) breakdown for all public Kitti odometry sequences, except 01. APE represents aligned trajectory absolute distance error, while RPE represents motion estimation error. On average, **32D-GL** provides the best results, with comparable performance from **32D-L**.

A visual comparison between the predicted trajectories for two Kitti odometry sequences can be found in Figure 7. As seen, the proposed method follows the ground truth more closely and presents less drift. In turn, this shows that our features are generally robust to revisitations and are viewpoint invariant.

Additionally, the average absolute and relative pose errors for the available Kitti sequences are shown in Table 5. These measures represent the absolute distance between the aligned trajectory poses and the error in the predicted motion, respectively. In this application, it can be seen how the system greatly benefits for the hierarchical aggregation learning approach. This is due to SLAM requiring two different sets of features. In order to estimate the motion of the agent in a narrow baseline, the system requires locally discriminative features. On the other hand, loop closure detection and map creation requires globally consistent features. This is refected in the results, where **G** consistently drifts more than **L** (higher RPE) and **GL** provides better absolute pose (lower APE).

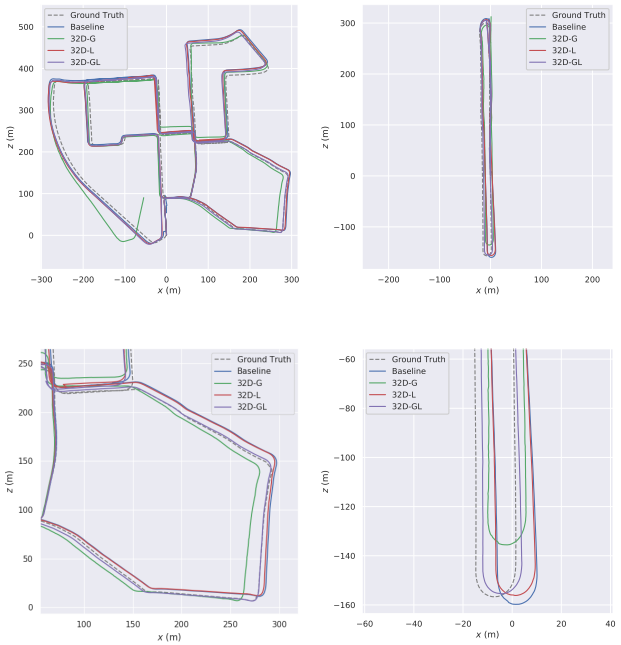


Figure 7: Kitti odometry trajectory predictions for varying SAND features vs. baseline. Top row shows two full sequences, with zoomed details in the bottom row. The hierarchical approach **GL** provides both robust motion and drift correction.

6. Conclusions & Future Work

We have presented SAND, a novel method for dense feature descriptor learning with a pixel-wise contrastive loss. By using sparsely labelled data from a fraction of the available training data we demonstrate that it is possible to learn generic feature representations. While other methods employ hard negative mining as a way to increase robustness, we instead develop a generic contrastive loss framework allowing us to modify and manipulate the learned feature space. This results in a hierarchical aggregation of contextual information visible to each pixel throughout training.

In order to demonstrate the generality and applicability of this approach, we evaluate it on a series of different computer vision applications each requiring different feature properties. This ranges from dense and sparse correlation detection to holistic image description and pixel-wise classification. In all cases SAND features are shown to outperform the original baselines.

We hope this is a useful tool for most areas of computer vision research by providing easier to use features requiring less or no training. Further work in this area could include exploring additional desirable properties for the learnt features spaces and the application of these to novel tasks. Additionally, in order to increase the generality of these features they can be trained with much larger datasets containing a larger variety of environments, such as indoor scenes or seasonal changes.

Acknowledgements

This work was funded by the EPSRC under grant agreement (EP/R512217/1). We would also like to thank NVIDIA Corporation for their Titan Xp GPU grant.

References

[1] H. Altwaijry, A. Veit, and S. Belongie. Learning to Detect and Match Keypoints with Deep Architectures. In *Proceedings of the British Machine Vision Conference 2016*, pages 49.1–49.12, 2016. 2

[2] H. Badino, A. Yamamoto, and T. Kanade. Visual odometry by multi-frame feature integration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 222–229. IEEE, dec 2013. 1

[3] A. Balasundaram and C. Chellappan. Vision Based Motion Tracking in Real Time Videos. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–4. IEEE, dec 2017. 1

[4] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-Aware Learning of Maps for Camera Localization. *CVPR*, dec 2018. 2

[5] J.-R. Chang and Y.-S. Chen. Pyramid Stereo Matching Network. 2018. 2, 5, 6

[6] L.-C. Chen, G. Papandreou, K. Murphy, and A. L. Yuille. SEMANTIC IMAGE SEGMENTATION WITH DEEP CON-VOLUTIONAL NETS AND FULLY CONNECTED CRFS. Technical report, 2015. 1

[7] S. Chopra, R. Hadsell, and L. Y. Learning a similiarty metric discriminatively, with application to face verification. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 349–356, 2005. 2

[8] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg. ECO: Efficient convolution operators for tracking. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 6931–6939, nov 2017. 2

[9] S. Dasiopoulou, E. Giannakidou, G. Litos, P. Malasioti, and I. Kompatsiaris. A Survey of Semantic Image and Video Annotation Tools. Technical report. 2

[10] M. E. Fathy, Q.-H. Tran, M. Z. Zia, P. Vernaza, and M. Chandraker. Hierarchical Metric Learning and Matching for 2D and 3D Geometric Correspondences. *ECCV*, 2018. 2

[11] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. DeepStereo: Learning to Predict New Views from the World's Imagery. *CVPR*, 2016. 5

[12] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Transactions on Robotics*, 33(2):249–265, apr 2017. 2

[13] W. Ge, W. Huang, D. Dong, and M. R. Scott. Deep Metric Learning with Hierarchical Triplet Loss. *ECCV2018*, page ECCV2018, 2018. 2

[14] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 4

[15] G. Georgakis, S. Karanam, Z. Wu, J. Ernst, and J. Kosecka. End-to-end learning of keypoint detector and descriptor for pose invariant 3D matching. 2018. 2

[16] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1735–1742, 2006. 1, 2

[17] A. E. Johnson, S. B. Goldberg, Y. Cheng, and L. H. Matthies. Robust and efficient stereo feature tracking for visual odometry. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 39–46. IEEE, may 2008. 1

[18] B.-N. Kang, Y. Kim, and D. Kim. Pairwise Relational Networks for Face Recognition. 2018. 2

[19] A. Kendall, M. Grimes, and R. Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2015 Inter, pages 2938–2946, 2015. 2, 7

[20] B. Kitt, A. Geiger, and H. Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 486–492. IEEE, jun 2010. 1

[21] K. Lenc and A. Vedaldi. Learning covariant feature detectors. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9915 LNCS, pages 100–117. Springer, Cham, 2016. 2

[22] R. Li, S. Wang, Z. Long, and D. Gu. UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7291, may 2018. 2

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June, pages 3431–3440, 2015. 2

[24] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[25] D. Martin, R. Andreas, F. S. Khan, and M. Felsberg. Beyond Correlation Filters: Learning Continuous Convolution Operator for Visual Tracking. *ECCV*, 2016. 2

[26] R. Mechrez, I. Talmi, and L. Zelnik-Manor. The Contextual Loss for Image Transformation with Non-Aligned Data. *ECCV*, mar 2018. 2

[27] R. Mur-Artal, J. M. Montiel, and J. D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, oct 2015. 2

[28] R. Mur-Artal and J. D. Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, oct 2017. 2

[29] T. Naseer and W. Burgard. Deep Regression for Monocular Camera-based 6-DoF Global Localization in Outdoor Environments. Technical report, 2017. 1

[30] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-Time. *Computer Vision and Pattern Recognition (CVPR)*, pages 343–352. 2

[31] H. Noh, S. Hong, and B. Han. Learning Deconvolution Network for Semantic Segmentation. 1:1520–1528, 2015. 2

[32] D. Patel and S. Upadhyay. Optical Flow Measurement using Lucas kanade Method. *International Journal of Computer Applications*, 61(10):975–8887, 2013. 1

[33] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. Jacobo Berlles. S-PTAM: Stereo Parallel Tracking and Mapping. *Robotics and Autonomous Systems*, 93:27–42, jul 2017. 2, 7

[34] R. a. N. Rse. KinectFusion : Real-Time Dense Surface Mapping and Tracking. Technical report, 2013. 2

[35] E. Rublee and G. Bradski. ORB: an efcient alternative to SIFT or SURF. Technical report, 2012. 2

[36] Y. Sakai, T. Oda, M. Ikeda, and L. Barolli. An object tracking system based on SIFT and SURF feature extraction methods. In *Proceedings - 2015 18th International Conference on Network-Based Information Systems, NBiS 2015*, pages 561–565. IEEE, sep 2015. 2

[37] S. Salti, F. Tombari, R. Spezialetti, and L. D. Stefano. Learning a descriptor-specific 3D keypoint detector. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2015 Inter, pages 2318–2326. IEEE, dec 2015. 2

[38] T. Schmidt, R. Newcombe, and D. Fox. Self-Supervised Visual Descriptor Learning for Dense Correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2017. 2, 3

[39] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. mar 2015. 2

[40] Y. Sun, X. Wang, and X. Tang. DeepID2: Deep Learning Face Representation by Joint Identification-Verification. *Nips*, pages 1–9, 2014. 2

[41] J. Wang, Y. Xing, and G. Zeng. Attention Forest for Semantic Segmentation. pages 550–561. Springer, Cham, nov 2018. 2

[42] S. Wang, R. Clark, H. Wen, and N. Trigoni. DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2043–2050, sep 2017. 2

[43] S. Wu, Y. Fan, S. Zheng, and H. Yang. Object tracking based on ORB and temporal-spacial constraint. In *2012 IEEE 5th International Conference on Advanced Computational Intelligence, ICACI 2012*, pages 597–600. IEEE, oct 2012. 2

[44] G. Yang and Z. Deng. End-to-End Disparity Estimation with Multi-granularity Fully Convolutional Network. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10636 LNCS, pages 238–248. Springer, Cham, nov 2017. 1

[45] B. Yu. Correcting the Triplet Selection Bias for Triplet Loss. In *ECCV*, 2018. 2

[46] J. Zhang and S. Singh. Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 1

[47] J. Zhang, K. A. Skinner, R. Vasudevan, and M. Johnson-Roberson. DispSegNet: Leveraging Semantics for End-to-End Learning of Disparity Estimation from Stereo Imagery. 2018. 2

[48] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-Janua, pages 6612–6621, 2017. 2