Moving SLAM: Fully Unsupervised Deep Learning in Non-Rigid Scenes

Dan Xu[1], Andrea Vedaldi[2], João F. Henriques[2]

Abstract— **We propose a new deep learning framework to decompose monocular videos into 3D geometry (camera pose and depth), moving objects, and their motions, with no supervision. We build upon the idea of view synthesis, which uses classical camera geometry to re-render a source image from a different point-of-view to obtain supervisory signals, specified by a predicted relative 6-degree-of-freedom pose and depth map. However, the typical view synthesis equations rely on a strong assumption: that objects in scenes do not move. This rigid-world assumption limits the predictive power, and rules out learning about objects automatically. We propose a simple solution: minimize the synthesis error on small local regions of the image instead. While the scene as a whole may be non-rigid, it is always possible to find small regions that are approximately rigid, such as inside a moving object. Our network can learn a *dense* pose map describing poses for each local region. This represents a significantly richer model, including 6D object motions, with little additional complexity. We establish very competitive results on unsupervised odometry and depth prediction on KITTI. We also demonstrate new capabilities on EPIC-Kitchens, a challenging dataset of indoor videos, where there is no ground truth information for depth, odometry, object segmentation or motion - yet all are recovered automatically by our approach.**

I. Introduction

It is a long-standing goal in both robotics and computer vision to achieve a holistic understanding of a visual scene: that is, to decompose it into meaningful elements that together explain the full visual input [1]. This goal is also the core of representation learning, which is concerned with extracting representations from data that generalize well for multiple tasks [2]. For example, a representation that was trained for object detection will invariably ignore details that are crucial for other tasks that are not object-centric, such as monocular depth estimation. As such, it is desirable to learn models that are not narrowly-scoped to a single task, although it is not always clear how to do so without an increased annotation burden for each additional task.

A recent line of work that promises to achieve both goals at once, i.e., a holistic scene understanding without demanding additional annotations, considering unsupervised learning via view synthesis [3], [4]. It cleverly combines two predictions (i.e. tasks), relative camera pose estimation and depth estimation, to re-render an image from another point-of-view. By synthesizing a source frame from a video into another target frame, we obtain the necessary supervisory signals (i.e. the error between the synthetic and the real image) to enable end-to-end learning.

It is natural to ask if more tasks can be integrated into this framework, and intuitively the answer is yes. Each additional task that contributes to the image synthesis, by capturing an additional element of the visual world, should increase the fidelity of the model and achieve a lower reconstruction error. In this work, we propose to add the unsupervised tasks of object segmentation, and object 6D motion estimation. Since our primary interest is still the recovery of 3D geometry and motion, however, we do not seek, as done by some prior works [5], [6], [7], “semantic” objects. Instead, we decompose the image into regions that are likely to be characterized by a well defined rigid motion, and learn those automatically, by optimizing for the same view synthesis objective. Our technical contribution is a locally-rigid model that supports crisp object boundaries through segmentation, and an efficient implementation of a “tiling” operator for local region reconstruction. This results in an effective lightweight model that transforms the original rigid-world model where changes between frames are characterized only by a single relative camera pose and depths [4], to a non-rigid model where moving objects are also taken into account.

Experiments show that our proposed model is much more expressive, by making predictions that are outside the capabilities of the original model (namely segment moving objects and calculate their 6D motions). We show qualitative results of our neural networks successfully learning to segment non-rigid objects (hands and household objects), and recover accurate depth maps, in EPIC-Kitchens [8], a large-scale indoors dataset that has not been used in this context before due to its challenging nature. We also demonstrate that the unsupervised segmentation cues and non-rigid model are beneficial for the previously-considered tasks. Our model achieves very competitive performance compared with state-of-the-art results on KITTI [9] in monocular depth estimation (88.9% accuracy at $\delta < 1.25$), as well as visual odometry (0.011/0.010 ATE), from purely unsupervised data.

II. Related Work

Self-supervised learning [10] has gathered significant attention recently, since it promises to achieve unsupervised learning by reusing standard elements from supervised learning (e.g. architectures, loss functions), in relatively intuitive configurations. It can generally be described as the task of predicting one part of the input data given only another part. Successful examples include predicting the spatial relationship between two image regions [11], colorizing images [12],

[1] Dan Xu is with Department of Computer Science and Engineering, Hong Kong University of Science and Technology, HK. Email: danxu@cse.ust.hk

[2] Andrea Vedaldi and João F. Henriques are with Visual Geometry Group (VGG), University of Oxford, UK. Email: {vedaldi, joao}@robots.ox.ac.uk

and predicting inertial measurement unit (IMU) data (odometry) [13]. Within self-supervised learning, video prediction is a relatively popular task. It consists of generating a subset of a video given the remaining video, such as video generation with neural networks [14], and causal convolutions as the PixelCNN for video [15]. While successful, these approaches often produce blurry or physically-implausible predictions, and the image generation models are not interpretable.

Approaching this problem by view synthesis [4] is relatively recent, but draws from traditional works in visual geometry and Simultaneous Location And Mapping (SLAM) [16], [17]. Instead of using a neural network for generation, it uses a differentiable warp (image deformation) to transform a source frame into a target frame, minimizing the reconstruction error. The task of the neural network is then to predict physically-interpretable quantities, such as 3D geometry, depth and poses, that are used to compute the image warp. This physical model of the world is useful in itself, and can be used for downstream tasks (such as visual odometry or 3D reconstruction). This line of work was pioneered by Garg et al. [3], who considered stereo pairs or pairs of frames with known pose. This was further developed to include predicted poses by Zhou et al. [4] (SfMLearner), and concurrently by Vijayanarasimhan et al. [18] (SfM-Net), who also considered multiple layers to account for objects. The SfMLearner emphasized no supervision (as opposed to mixed modes of supervision like SfM-Net), and a simple architecture, so it forms the basis of our work. We describe it in detail in Sec. III. Further developments include improving the pose estimate by a direct visual odometry method (essentially second-order gradient descent) [19], and adding a FlowNet that can refine the image warp by fine-grained optical flow estimates [20]. Other proposed improvements are stereo inputs [21], [22] and probabilistic outputs [23], bundle adjustment to warp stored key frames instead of recent frames [24], adversarial training [25], and feature computation in 3D space [26].

III. Unsupervised learning by view synthesis

We begin by describing the canonical self-supervised setup for learning monocular depth estimation, inspired by early methods such as SfMLearner [4] (Sec. II). Assume that we are given a pair of images $(I^{\mathrm{s}}, I^{\mathrm{t}})$, respectively the *source* and *target*, usually extracted as nearby frames from a video. If the scene is Lambertian and unchanged between frames (rigid-world assumption) and if we discount occlusions, the target image $\hat{I}^{\mathrm{t}}$ can be predicted from knowledge of the source image I^{s}, the depth map D, and the relative camera pose P between the two views:

$$\hat{I}^{\mathrm{t}} = \Psi\left(I^{\mathrm{s}}, P, D\right) \tag{1}$$

Synthesizing the target image amounts to using projective geometry [1] to find which pixels correspond in the two views and then transport their intensities from source to target. Namely, the synthesis function Ψ is a warp that linearly interpolates each pixel of I^{s} at coordinates (u,v) to (u',v') according to the projective equations:

$$\left[u',v',1\right]^T = KPZ\left(D_{uv}\right)K^{-1}\left[u,v,1\right]^T, \tag{2}$$

where K is the camera intrinsics matrix, D_{uv} retrieves the depth at pixels (u,v), and $Z(D_{uv})$ is the transformation matrix that translates any input 3D point along the z axis by this depth. We can extract a supervisory signal for depth and pose by comparing the measured and predicted target image:

$$L_{\mathrm{rigid}} = \frac{1}{|\mathcal{D}|} \sum_{(I^{\mathrm{s}},I^{\mathrm{t}})\in\mathcal{D}} \left|I^{\mathrm{t}} - \Psi\left(I^{\mathrm{s}}, P, D\right)\right|_M \tag{3}$$

where $\mathcal{D}$ is a training dataset of image pairs, M is a mask that tells which pixels can be explained by the model and $|v|_M = \sum_k M_k |v_k|$, i.e. the L^1 norm weighted by a mask M.

Of course, the pose P, depth D and mask M must be obtained somehow. This is the role of the pose, depth and mask networks:

$$P = \rho\left(I^{\mathrm{t}}, I^{\mathrm{s}}|\omega\right), \quad d = \phi\left(I^{\mathrm{t}}|\omega\right), \quad M = \varepsilon\left(I^{\mathrm{t}}, I^{\mathrm{s}}|\omega\right) \tag{4}$$

with parameters ω. The last layer of the mask network is a sigmoid, to constrain outputs to the $[0,1]$ range. Backpropagating the error in Eq. 3 to the networks' parameters ω is what allows end-to-end learning. Since predicting $M = 0$ is a trivial minimum of Eq. 3, a regularization term is added, penalizing the distance between M and a target of 1:

$$L_{\mathrm{reg}} = \|M - 1\|^2 \tag{5}$$

The overall objective is then $\min_\omega L_{\mathrm{rigid}} + \lambda L_{\mathrm{reg}}$, with a regularization weight λ.

IV. Proposed method

As we mentioned in Sec. I, the main limitation of the SfM-Learner method from Sec. III is that it assumes a rigid world. Our main goal then is to augment it to also account for freely-moving, potentially non-rigid objects. Note that since most pixels in common scenes correspond to a static background, it is undesirable to have a fully non-rigid model, as that would afford too many degrees-of-freedom and thus be prone to overfitting (a hypothesis that we verify in Sec. V). For this reason, we segment the pixels into 2 categories: background (with a global rigid model), and objects (with a local, non-rigid model). Together, they fully explain all pixels of the image. The required segmentation network is trained unsupervised as part of the overall objective, and so object segmentation is obtained "for free". An overview of the framework is illustrated in Fig. 1. We will now describe these three elements: non-rigid and rigid models, and segmentation.

A. Locally-rigid scene model

The starting point for our method is the simple fact that, although a scene is not globally rigid, it usually is *locally* rigid. It is always possible to make the rigid-world assumption essentially correct, by narrowing down the view to a rigid region (e.g. background, a rigid object, or a smaller region within a non-rigid object). This seems to suggest that the previous method (Sec. III) can express a non-rigid

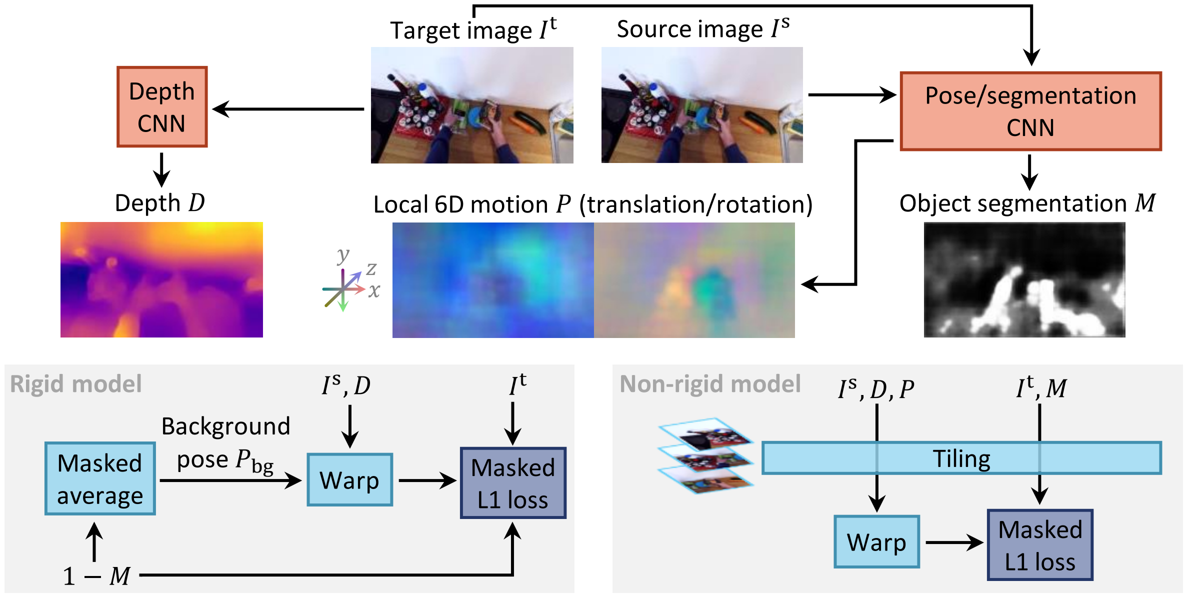


Fig. 1: Overview. **Top panel:** From a pair of video frames, CNNs predict 3 maps: depth, local *dense* 6D motion, and foreground segmentation. Higher depths are darker; the axes of motion translation and rotation (XYZ) are encoded as RGB channels; and foreground segmentation is white. **Bottom-left:** We model background pixels as a rigid object, with global 6D motion obtained by averaging their predicted motions. An L1 reconstruction error w.r.t. target image I^{t} allows learning. **Bottom-right:** Foreground pixels are similar, but only *locally-rigid*; inputs are passed through a *tiling* operator, dividing them into small patches. This allows motions per object/patch, unlike the fully rigid model.

model, by focusing on smaller regions of the image at a time instead of full images. We can then predict different relative poses for each local region, instead of a single pose P for the whole dynamic scene. We thus propose to average the objective (Eq. 3) over a set of regions $\mathcal{R}$, extracted with sliding windows:

$$L_{\text{non-rigid}} = \frac{1}{|\mathcal{D}||\mathcal{R}|} \sum_{(I^{\mathrm{s}},I^{\mathrm{t}})\in\mathcal{D}} \sum_{r\in\mathcal{R}} \left| I^{\mathrm{t}} - \Psi(I^{\mathrm{s}}, P_r, D) \right|_{M\cdot\mathbf{1}_r}, \quad (6)$$

where we element-wise multiply $(\cdot)$ the mask M with an indicator function $\mathbf{1}_r$, which is 1 inside the region r and 0 outside of it. P_r denotes the corresponding pose, extracted from a pose map P at the center of the region. The dense map of pose predictions P fits naturally into the CNN-based architecture, in the same way as the depth predictions D.

While it achieves our goal of a locally-rigid model, Eq. 6 is very inefficient (by summing over many zeros), and an efficient implementation would require customizing the differentiable warp operator Ψ. However, we can ensure an easy implementation with no modification to Ψ by using an operator that extracts patches in a sliding window, and concatenates them as samples before they are passed on to Ψ. This *tiling* operator, also known as `im2row`, is used in some implementations of convolutions. We can write it succinctly, for an input with c channels and spatial dimensions (w,h), as $t : \mathbb{R}^{c\times w\times h} \mapsto \mathbb{R}^{b\times c\times k\times k}$:

$$t_{i+jw,:,:,:}(X) = X_{:,si:si+k,sj:sj+k} \quad (7)$$

where $:$ is the tensor slice operator, the patches are $k\times k$, and the stride is s. Eq. 6 then becomes:

$$\begin{aligned} L_{\text{non-rigid}} &= L_{\text{rigid}}\left(\mathcal{D}', \operatorname{vec}(P), t(D), t(M)\right), \\ \mathcal{D}' &= \left\{\left(t(I^{\mathrm{s}}), t(I^{\mathrm{t}})\right) : (I^{\mathrm{s}}, I^{\mathrm{t}}) \in \mathcal{D}\right\}, \end{aligned} \quad (8)$$

which effectively amounts to applying the tiling operator t to all image-sized inputs of the original rigid-model objective (Eq. 3), and vectorizing the pose map $P \in \mathbb{R}^{6\times w\times h}$ so that the spatial dimensions correspond to the batch dimension ($\operatorname{vec}(P) \in \mathbb{R}^{wh\times 6}$). Since the kernel size k is unknown, in practice we repeat the objective for several values of k, corresponding to different object sizes. Our proposed inclusion of the tiling operators is illustrated in Fig. 1.

B. Object segmentation

As discussed in Sec. IV, a fully non-rigid model contains too many degrees-of-freedom. Another issue is that regions defined by square sliding windows (Eq. 6) are too coarse to accurately delineate objects' boundaries. We propose to solve both issues at once by using the predicted mask M to partition the pixels into moving objects ($M \simeq 1$) and static background ($M \simeq 0$):

$$\begin{aligned} L_{\text{segm}} &= L_{\text{non-rigid}} + L_{\text{rigid}}(\mathcal{D}, P_{\text{bg}}, D, 1-M), \\ P_{\text{bg}} &= \tfrac{1}{wh} \textstyle\sum_{ij}^{wh} (1 - M_{ij}) P_{ij}, \end{aligned} \quad (9)$$

where P_{bg} is the background pose, obtained by averaging the pose of all background pixels. Unfortunately, since the non-rigid model is more expressive than the rigid one, assigning all pixels to the former will always attain a lower error. To prevent this trivial solution, we modify the regularization term (Eq. 3) to encourage a constant area of foreground pixels in a training batch. We consider that the top 10% of

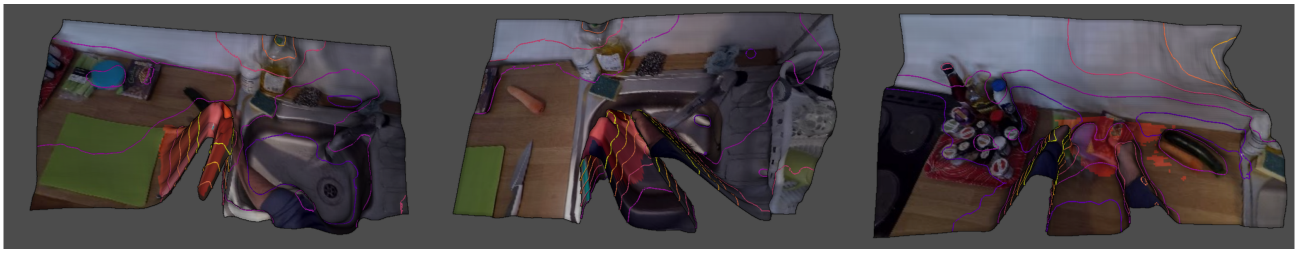


Fig. 2: Predicted 3D meshes and segmentations on the EPIC-Kitchens test set with our proposed unsupervised model. Contours of constant height are shown as lines (higher values are brighter). Note the correct geometry of the sink (left panel) and bottle cap (right panel). A failure mode is also visible on the right panel (back wall is distorted in the top-right). Object masks are tinted red; normally corresponding to moving hands or manipulated objects.

the predictions M correspond to pixels with moving objects ($M \simeq 1$), and the rest to background ($M \simeq 0$), penalizing the distance to these target values:

$$L_{\text{reg'}} = \|\text{sort}(\text{vec}(M)) - u\|^2 \tag{10}$$

where sort operates in descending order, and u is a vector where the first $\rho\%$ of the elements are 1 and the rest are 0. This "constant area" soft constraint is inspired by a similar approach used for visualizing salient regions in deep network interpretability [27], and we found it to be an effective strategy to ensure a correct proportion of object and background pixels. Our overall objective is then $\min_\omega L_{\text{segm}} + \lambda L_{\text{reg'}}$, in analogy to Sec. III as also elaborated in Fig. 1.

V. Experiments

We conducted extensive experiments to validate our approach. The tasks we focused on were visual odometry, monocular depth estimation, and 6D motion segmentation.

A. Experimental Setup.

Datasets. We evaluate the proposed approach on two different large-scale datasets. The first one is the challenging autonomous driving benchmark **KITTI** [9]. For the depth estimation, we use the training split defined by Eigen et al. [28]. For evaluating the performance of the visual odometry, we use the KITTI Odometry dataset, training on sequences $00-08$ and testing on $09-10$. For the second dataset, we used **EPIC-Kitchens** [8], which is collected under various indoor kitchen scenarios, and is the largest dataset for egocentric vision. It contains 32 kitchens crossing 4 cities, totalling 55 hours of video. It captures rich non-rigid dynamic motions. Some examples are shown in Fig. 5. As the original frame rate of the videos is 60 FPS, to reduce the redundancy, we sample the dataset at every 4 frames, resulting in a dataset of around 120k images. Among them, 100k images are used for training, and the rest for testing. The dataset does not provide ground-truth depth, camera poses and intrinsics. We learn to recover all of them by unsupervised end-to-end learning.

Training setup. Our training procedure is exactly the same as for the SfMLearner [4], except with the non-rigid model we propose (Sec. IV). We used a ResNet-50 as the backbone CNN, and apply the same TV-norm (Sec. III) to both depth (D) and pose (P) maps. The only other improvements are depth mean normalizaton and backbone initialization according to Gordon et al. [29], as well as disabling the multi-scale prediction, which does not seem beneficial.

B. Overall Performance.

Pose estimation performance. Our sequential pose estimation (visual odometry) performance is reported in table I. We compare our method with a traditional monocular SLAM system, ORB-SLAM (full) [35], as well as its local version, ORB-SLAM (short), which uses 5-frame snippets. We also compare with the SfMLearner [4] and several recent proposals [30], [32], [20], [31], [23], [34]. As can be seen in table I, our method outperforms all the other methods. This includes a traditional SLAM pipeline that draws from many years of careful engineering and manual tuning (ORB-SLAM). It is worth mentioning that our method is not explicitly trained for visual odometry, yet it is a useful by-product of training. Regarding the deep learning based approaches, our camera motion estimator outperforms the original SfMLearner [4] by a large margin, and recent competing approaches by narrower but still significant margins. We include the standard deviations over several runs in table I for additional context.

Qualitative results on visual odometry. We visualize the predicted camera trajectories in Fig. 4a and 4b, for several algorithms, on two KITTI test sequences (09 and 10, respectively). All trajectories are registered w.r.t. the ground truth as standard [36]. It is apparent that the trajectories predicted by our method very accurately follows the ground truth. While other methods can also get close to the ground truth trajectory (e.g. SC-SFM in Fig. 4a and ORB-SLAM in Fig. 4b), no other can achieve the same accurate performance simultaneously on both settings.

Depth estimation performance. We report the result for depth estimation in table II. The columns are relative error and its square, root-mean-squared error (RMSE) and its logarithm, as well as the accuracy at 3 given depth thresholds. We include several state-of-the-art approaches in the comparison, including supervised methods [28], [37] and unsupervised stereo-based methods [21], [3], which are not comparable but present an informative upper bound on performance. We do not include in the comparison works with significantly different protocols, such as 3DPackNet [26] which uses higher-resolution images and 3D CNNs, and other works with higher amounts of supervision. These interesting de-

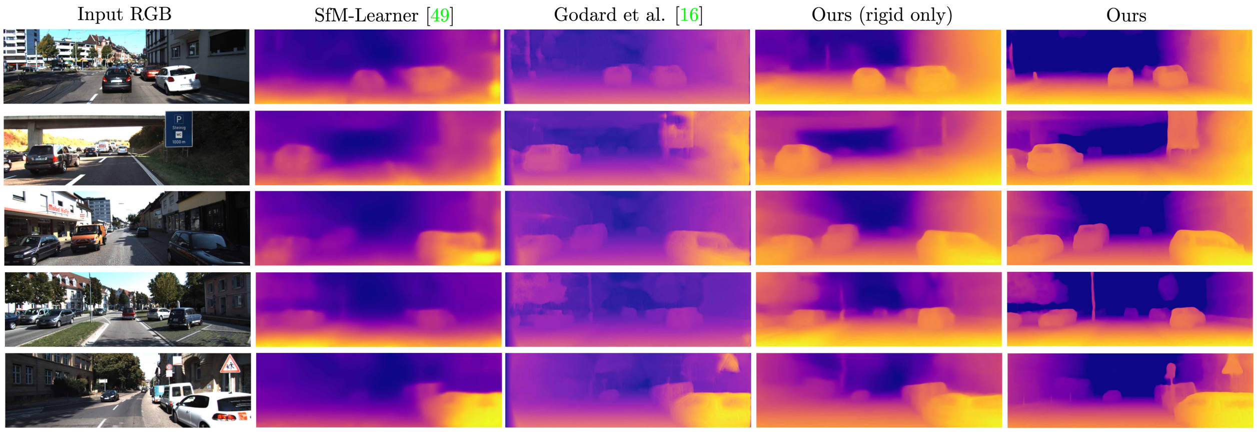
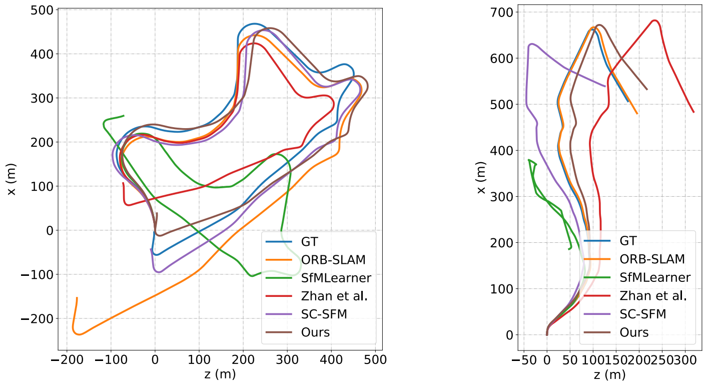


Fig. 3: Visualisation of depth estimates for our method, including an ablation without the non-rigid component, and two other methods, including one with access to stereo information [21]. Our method can accurately capture moving cars and fine details such as traffic signs and poles.

Method	Absolute Trajectory Error	
	sequence 09	sequence 10
Mean Odometry	0.032 ± 0.026	0.028 ± 0.023
ORB-SLAM (short)	0.064 ± 0.141	0.064 ± 0.130
ORB-SLAM (full)	0.014 ± 0.008	0.012 ± 0.011
Zhou et al. [4]	0.021 ± 0.017	0.020 ± 0.015
DF-Net [30]	0.017 ± 0.007	0.015 ± 0.009
Monodepth2 [29]	0.017 ± 0.008	0.015 ± 0.010
Zhou et al. [4] (new)	0.016 ± 0.009	0.013 ± 0.009
Bian et al. [31]	0.016 ± 0.007	0.015 ± 0.015
Klodt et al. [23]	0.014 ± 0.007	0.013 ± 0.009
Mahjourian et al. [32]	0.013 ± 0.010	0.012 ± 0.011
EPC++ [33]	0.013 ± 0.007	0.012 ± 0.008
GeoNet [20]	0.012 ± 0.007	0.012 ± 0.009
CC [34]	0.012 ± 0.007	0.012 ± 0.009
Ours	**0.011** ± **0.005**	**0.010** ± **0.007**

TABLE I: Absolute Trajectory Error (ATE) on the KITTI odometry test split averaged over all 5-frame snippets. Our approach outperforms all others, including a traditional SLAM pipeline.



(a) Testing sequence 09

(b) Testing sequence 10

Fig. 4: Qualitative state-of-the-art comparison of the visual odometry results on the full testing sequences 09 and 10 of the KITTI odometry dataset.

velopments are complementary to ours, and thus outside of the scope of this paper. All methods are compared with raw ground truth depth, following recent protocols [29], [26]. We can observe that our method achieves the best performance out of all unsupervised methods, even beating several supervised ones, on almost all metrics. We show two variants of our method, using ResNet-18 and ResNet-50 backbone CNNs, which highlights that the performance gains are not solely due to an increase in capacity, since most methods have comparable backbones. On the other hand, it also reveals that our model is expressive enough to afford some gains when increasing the capacity to ResNet-50. More importantly, our method achieves clearly better results compared with two recent works, i.e. Struct2Depth [5] and Tosi et al. [6], which use explicit semantic labels to guide the learning of object motion, demonstrating the benefits of unsupervised segmentation.

C. Ablation Study and Qualitative Results.

Ablative study. Although our model adds relatively few elements to view synthesis, it is important to know their relative impact on performance. We show the performance of our system with the addition of each element in tables III-IV. It is apparent that the biggest boost comes from adding our non-rigid model, especially in visual odometry performance. We also show a similar breakdown, but *adding* our improvements to the original SfMLearner system [4], which has a weaker backbone CNN and no depth normalization. This shows that the benefits of our non-rigid model are complementary to other improvements, yet very significant.

Qualitative results on depth. Fig. 3 shows a direct comparison of the depth estimates from our method, the most comparable baseline by Zhou et al. [4], and a stereo-based method [3]. Our method recovers much finer details, compared to a rigid-world model (4th column). It is also interesting to compare the level of detail with the stereo method (3rd column), which achieves better error metrics (table II), but has relatively inconsistent fine details. Ours seems to strike a good balance between capturing the high-level layout and small-scale features. We also show qualitative results of our method in EPIC-Kitchens, for which there is no ground truth, in Fig. 5. Despite the challenges of the quick camera and

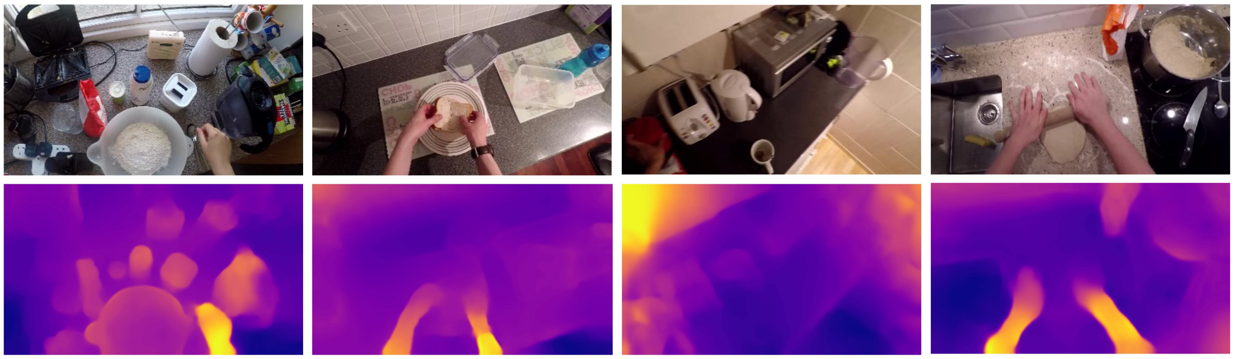


Fig. 5: Qualitative examples of our unsupervised depth estimation on the EPIC-Kitchens dataset. Despite the fast, non-rigid motions, we can recover detailed structures, such as the tabletop objects.

TABLE II: Quantitative comparison of depth estimation performance among several methods from the literature, on the KITTI raw dataset (Eigen et al. [28] testing split). We show some supervised (denoted 'D') and stereo ('S') methods for reference, but a fair comparison is only with monocular methods ('M') trained with video ('V'). Our method outperforms others in most metrics, even when using a ResNet-18 backbone with much fewer parameters.

Method	Setting	Error (lower is better)				Accuracy (higher is better)		
		rel	sq rel	rmse	rmse (log)	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al. [28]	M + D	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [37]	M + D	0.202	1.614	6.523	0.275	0.678	0.895	0.965
MS-CRF [38]	M + D	0.125	0.899	4.685	-	0.816	0.951	0.983
AdaDepth [39]	S	0.203	1.734	6.251	0.284	0.687	0.899	0.958
Garg et al. [3]	S	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard et al. et al. [21]	S	0.124	1.076	5.311	0.219	0.847	0.942	0.973
Kuznietsov et al. [40]	S + D	0.113	0.741	4.621	0.189	0.862	0.960	0.986
Zhou et al. [4]	M + V	0.208	1.768	6.858	0.283	0.678	0.885	0.957
Yang et al. [41]	M + V	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian [32]	M + V	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Geonet (ResNet) [20]	M + V	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Wang et al. [42]	M + V	0.151	1.257	5.583	0.228	0.810	0.936	0.974
DF-Net [30]	M + V	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Struct2Depth [5]	M + V	0.141	1.026	5.291	0.215	0.816	0.945	0.979
CC (ResNet) [34]	M + V	0.140	1.070	5.326	0.217	0.826	0.941	0.975
Bian et al. [31]	M + V	0.128	1.047	5.234	0.208	0.846	0.947	0.976
Gordon et al. [29]	M + V	0.128	0.959	5.230	–	–	–	–
Tosi et al. [6]	M + V	0.125	**0.805**	4.795	0.195	0.849	0.955	0.983
Monodepth2 [43]	M + V	0.115	0.882	**4.701**	0.190	0.879	0.961	0.982
Ours (ResNet-18)	M + V	0.105	0.889	4.780	0.182	0.884	0.961	0.982
Ours (ResNet-50)	M + V	**0.103**	0.881	4.763	**0.179**	**0.889**	**0.964**	**0.984**

object motions in this setting, the quality of the recovered depths is apparent, making it possible to identify individual small objects and fine geometry.

Qualitative results on object discovery and motion prediction. Since our proposed method can predict relative poses (6D motion) densely for the whole image and separate foreground pixels (with independent motions) from static background pixels (with a single coherent motion), it should be able to discover moving objects in the image automatically. We validate this idea by binarizing the mask M (with a threshold of 0.7 and $\rho\%$ as 10%), and overlaying it on the original data. The resulting segmentations in EPIC-Kitchens can be visualized in Fig. 2, along with the projected 3D geometry. We can observe the correct segmentation and clustering of the hands, and of objects as they are being held. We show more visualizations of the motion prediction in the supplementary video material.

VI. Conclusion

We have proposed a simple approach to view synthesis for self-supervised learning, which leverages a successful rigid-world model and augments it with a locally-rigid model instead. This is done by strategic placement of tiling operators within the network, which is both efficient and produces highly consistent depth and pose estimates. Our change enables unsupervised object discovery by motion segmentation, and allows associating 3D pose and 6D motion to different objects in the scene. We also demonstrated very competitive performance in unsupervised visual odometry and monocular depth estimation compared with state-of-the-art results. We hope that our contribution enables a new class of non-rigid SLAM algorithms, that continue the trend towards truly holistic scene understanding.

Acknowledgement: The authors thank the generous support

Method	rel	sq rel	rmse
SfMLearner [4]	0.208	1.768	6.858
SfMLearner + our non-rigid model	0.173	1.302	6.252
+ regularization (TV-norm)	**0.165**	**1.301**	**6.225**
Our rigid model (Eq. 3)	0.121	1.115	5.285
Our non-rigid model (Eq. 8)	0.108	0.899	4.788
+ regularization (TV-norm)	0.107	0.895	4.786
+ segmentation (Eq. 9-10) (full system)	**0.105**	**0.889**	**4.780**

TABLE III: Ablation study on monocular depth estimation (error). See Sec. V for details.

Method	seq. 09 (ATE)	seq. 10 (ATE)
SfMLearner[4]	0.021 ± 0.017	0.020 ± 0.015
SfMLearner + our non-rigid model	0.012 ± 0.007	0.011 ± 0.008
+ regularization (TV-norm)	**0.011** ± **0.005**	**0.011** ± **0.007**
Our rigid model (Eq. 3)	0.016 ± 0.012	0.015 ± 0.013
Our non-rigid model (Eq. 8)	0.013 ± 0.006	0.011 ± 0.010
+ regularization (TV-norm)	0.012 ± 0.006	0.011 ± 0.009
+ segmentation (Eq. 9-10) (full system)	**0.011** ± **0.005**	**0.010** ± **0.007**

TABLE IV: Ablation study on visual odometry (Absolute Trajectory Error and std. dev.). See Sec. V for details.

of the Royal Academy of Engineering (RF\201819\18\163), the EPSRC Programme Grant Seebibyte EP/M013774/1, the ERC 638009-IDIU, and of Continental Corporation. The work is also supported in part by the Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No.26202321 and HKUST Startup Fund No. R9253.

References

[1] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[2] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *CVPR*, 2018.

[3] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *ECCV*, 2016.

[4] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, 2017.

[5] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *AAAI*, 2019.

[6] F. Tosi, F. Aleotti, P. Z. Ramirez, M. Poggi, S. Salti, L. Di Stefano, and S. Mattoccia, "Distilled semantics for comprehensive scene understanding from videos," in *CVPR*, 2020.

[7] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically-guided representation learning for self-supervised monocular depth," in *ICLR*, 2020.

[8] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "The epic-kitchens dataset: Collection, challenges and baselines," *TPAMI*, 2020.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *IJRR*, 2013.

[10] R. Caruana and V. R. De Sa, "Promoting poor features to supervisors: Some inputs work better as outputs," in *NIPS*, 1997.

[11] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *ICCV*, 2015.

[12] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *ECCV*, 2016.

[13] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *ICCV*, 2015.

[14] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *ICML*, 2015.

[15] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Video pixel networks," in *ICML*, 2017.

[16] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[17] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *TPAMI*, no. 6, pp. 1052–1067, 2007.

[18] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *CoRR*, vol. abs/1704.07804, 2017. [Online]. Available: http://arxiv.org/abs/1704.07804

[19] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *CVPR*, 2018.

[20] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *CVPR*, 2018.

[21] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[22] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," in *ICRA*, 2018.

[23] M. Klodt and A. Vedaldi, "Supervising the new with the old: learning SFM from SFM," in *ECCV*, 2018.

[24] N. Yang, R. Wang, J. Stückler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *ECCV*, 2018.

[25] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe, "Unsupervised adversarial depth estimation using cycled generative networks," in *3DV*, 2018.

[26] V. Guizilini, R. Ambrus, S. Pillai, and A. Gaidon, "3d packing for self-supervised monocular depth estimation," in *CVPR*, 2020.

[27] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," in *ICCV*, 2019.

[28] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014.

[29] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *ICCV*, 2019.

[30] Y. Zou, Z. Luo, and J.-B. Huang, "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency," in *ECCV*, 2018.

[31] J.-W. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *NeurIPS*, 2019.

[32] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *CVPR*, 2018.

[33] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *TPAMI*, 2019.

[34] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *CVPR*, 2019.

[35] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *TRO*, vol. 31, no. 5, pp. 1147–1163, 2015.

[36] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *CVPR*, 2018.

[37] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *CVPR*, 2015.

[38] D. Xu, E. Ricci, W. Ouyang, X. Wang, N. Sebe, *et al.*, "Monocular depth estimation using multi-scale continuous crfs as sequential deep networks," *TPAMI*, vol. 41, no. 6, pp. 1426–1440, 2018.

[39] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu, "Adadepth: Unsupervised content congruent adaptation for depth estimation," in *CVPR*, 2018.

[40] Y. Kuznietsov, J. Stückler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *CVPR*, 2017.

[41] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia, "Unsupervised learning of geometry with edge-aware depth-normal consistency," in *AAAI*, 2018.

[42] C. Wang, J. M. Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *CVPR*, 2018.

[43] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *ICCV*, 2019.