SalientDSO: Bringing Attention to Direct Sparse Odometry

Huai-Jen Liang, Nitin J. Sanket, Cornelia Fermüller, and Yiannis Aloimonos

***Abstract*— Although cluttered indoor scenes have a lot of useful high-level semantic information which can be used for mapping and localization, most visual odometry (VO) algorithms rely on the usage of geometric features such as points, lines, and planes. Lately, driven by this idea, the joint optimization of semantic labels and estimating odometry has gained popularity in the robotics community. This joint optimization method is accurate but is generally very slow. At the same time, in the vision community, direct and sparse approaches for VO have stricken the right balance between speed and accuracy. We merge the successes of these two communities and present a preprocessing method to incorporate semantic information in the form of visual saliency to direct sparse odometry (DSO)—a highly successful direct sparse VO algorithm. We also present a framework to filter the visual saliency based on scene parsing. Our framework *SalientDSO* relies on the widely successful deep learning-based approaches for visual saliency and scene parsing, which drives the feature selection for obtaining highly accurate and robust VO even in the presence of as few as 40 point features per frame. We provide an extensive quantitative evaluation of SalientDSO on the ICL-NUIM and the TUM monoVO data sets and show that we outperform DSO and ORB-simultaneous localization and mapping—two very popular state-of-the-art approaches in the literature. We also collect and publicly release a CVL-UMD data set which contains two indoor cluttered sequences on which we show qualitative evaluations. To the best of our knowledge, this is the first paper to use visual saliency and scene parsing to drive the feature selection in direct VO.**

***Note to Practitioners*—The algorithm of estimating the camera motion from a set of moving camera frames/images is commonly called VO. This problem has many applications ranging from building a 3-D map of the scene for the robot to navigate, grasp, and so on. Any VO algorithm must be fast, robust, and with low drift (low accumulation in error). These desired functions are generally obtained by selecting "good" features in an image, which, in the computer vision sense, turns out to be "corners." However, when we constrain the setting to an indoor scene with a lot of clutter, we have a lot of objects which can be used to obtain "good" features from both a computer vision sense and a conceptual sense. We use this philosophy and present a preprocessing method to select better features as compared to a traditional VO pipeline using only geometric features and improve the robustness of the state-of-the-art VO method: direct sparse odometry, obtaining more accurate and robust results even with the lesser number of features. We evaluate our methods on three different data sets: ICL-NUIM, TUM monoVO, and CVL-UMD. We collected a custom dataset we call CVL-UMD to demonstrate the robustness of our approach, namely, SalientDSO in cluttered indoor scenes.**

***Index Terms*— Direct sparse odometry (DSO), scene parsing, SLAM, visual saliency.**

Manuscript received September 7, 2018; revised December 11, 2018; accepted January 20, 2019. This paper was recommended for publication by Associate Editor H. Liu and Editor H. Liu upon evaluation of the reviewers' comments. This work was supported in part by the Office of Naval Research under Grant N00014-17-1-2622, in part by the National Science Foundation under Grant 1824198, and in part by Brin Family Foundation through a gift to the Perception and Robotics Group at the University of Maryland and the Northrop Graumman Corporation. *(Huai-Jen Liang and Nitin J. Sanket contributed equally to this work.) (Corresponding author: Nitin J. Sanket.)*

The authors are with the Institute for Advanced Computer Studies, University of Maryland at College Park, College Park, MD 20742 USA (e-mail: hjliang@terpmail.umd.edu; nitin@umiacs.umd.edu; fer@umiacs.umd.edu; yiannis@umiacs.umd.edu).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. It contains a video "SalientDSOVideo.mp4" that is a companion multimedia file demonstrating the key idea used in our proposed method. It shows the outputs of multiple experimental runs. The code to replicate results can be found at http://prg.cs.umd.edu/SalientDSO.html.

Color versions of one or more of the figures in this paper are available online at http:ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2019.2900980

I. Introduction and Philosophy

SIMULTANEOUS localization and mapping (SLAM) and visual odometry (VO) algorithms have taken center stage in recent years due to their widespread usage. They play a prominent part in the perception and planning pipelines of self-driving cars, autonomous quadrotors, augmented, and virtual reality. The never-ending quest to come up with real-time solutions for these methods while being as accurate as their offline counterparts has led to alternative problem formulations in terms of constraints and optimization methods [1]–[4].

Recently, the field was dominated by indirect methods [1], [2], [5], [6] which rely on feature matching and foundations of multiview geometry coupled with windowed optimization to build a map of the scene and obtain accurate poses. These approaches are based on the low-level geometric features and do not work very well with environments with repeating structures and/or textureless surfaces. Some works have improved upon the previous approaches in terms of speed and accuracy by incorporating prior knowledge such as the dynamics of the system and/or data from more sensors such as inertial measurement units [7], time-of-flight sensors [8], and so on. However, minimalism is a trend-forward, i.e., trying to achieve the same tasks with a minimal number of sensors [9]. In the scope of this paper, we focus on a monocular VO solution. The current state of the art in monocular approaches which have the best compromise of speed and accuracy are direct sparse approaches, such as direct sparse odometry (DSO) [10].

However, object-centric SLAM approaches are more robust by nature due to the high-level semantics used in their formulation. Lately, the joint optimization of 3-D poses, structure, and labeled object locations has improved the state-of-the-art significantly [11]. These frameworks rely on the widely

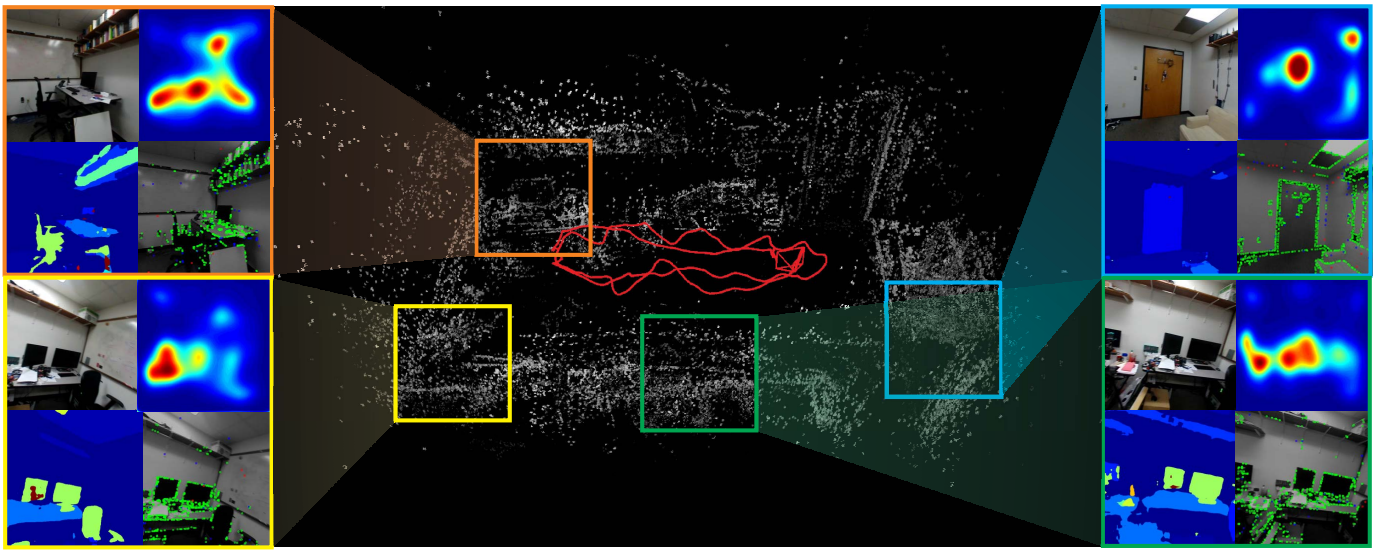

Fig. 1. Sample point-cloud output of SalientDSO which does not have loop closure or global bundle adjustment. Each inset (color-coded to suit the respective location on the map) in clockwise direction from top-left show the corresponding image, saliency, scene-parsing outputs, and active features. Observe that features from noninformative regions are almost removed approaching object-centric odometry. *All the images in this paper are best viewed in color.*

TABLE I
ACTIVE VERSUS PASSIVE APPROACH FOR COMPUTER VISION TASKS

Task	Passive approach	Active approach
Segmentation	Graph cut or super-pixel based methods.	Fixation based region segmentation and recognition in a feedback loop.
Recognition	Sliding window of filter banks with a classification algorithm for final prediction.	Saliency/fixation based segmentation/clustering followed by selection of attributes and sliding window of filters with a simple classification algorithm.
Tracking and Failure recovery	Making an online dictionary for robustness against changes and use detection for failure recovery.	Tightly couple saliency into the tracking filter to reduce search space and use salient regions for failure recovery. By doing so, we introduce high level semantics into the low level processes (feedback).
Navigation and Mapping	Map based on features selected on image gradients.	Map only using salient region features or objects obtained using fixation based segmentation. Take advantage of the semantic relationships between differently labeled regions.

successful deep learning-based object recognition engine and pose graph optimization frameworks, combining both the low-level geometric features and the high-level semantics.

However, humans perform the task of mapping very differently. The human visual system interprets the scene for various tasks such as recognition [12], segmentation [13], tracking [14], and navigation by making a series of fixations [15]. This is called the active approach [9], [16]–[18], whereas the traditional approach is called the passive approach (See Table I). These fixations lie in the protosegmentation of the salient objects/locations in the scene. The word protosegmentation refers to the fact that a segmentation around the fixation point may lead to partial/complete segmentation of an object, which depends on the scenario. Solving the problem of recognition and tracking along with segmentation is like a chicken-egg problem. One would need a good segmentation for recognition and tracking and vice versa. An expectation–maximization type of scheme, where one would jointly/alternatively optimize for the segmentation and recognition/tracking, has gained popularity in the literature lately due to the advancement of fast and accurate optimization frameworks.

At the same time, very recently, this philosophy of fixation and attention has started to gain popularity in the robot navigation community [11], [19]–[21]. This is based on the fact that humans perform the task of mapping very differently from how it has been done in the robotics literature. They build "semantic/topological" maps to traverse the scene. This paper combines the concepts used by the humans and robotics literature to present a framework of indoor VO in which the features are selected based on a visual saliency map that is obtained by human eye-tracking data. Our work is consistent with major theoretical works in the field of active perception [18], [22]. In [18], it is suggested that attention mechanisms precede any visual computation, and in [22], solutions to active perception problems may come from multimodal fusion. Indeed, our approach can be generalized to different inputs (LIDAR, radar, tactile, audio, etc.).

This paper aims to mimic the qualitative human vision in the framework of direct VO. The key contributions of this paper are as follows.

1) We present a framework of indoor VO in which the features are selected based on a visual saliency map (sample output is shown in Fig. 1).
2) We present a method to filter saliency map based on scene parsing.
3) We provide experimental results on various simulated and real indoor environments to demonstrate the improved performance of the proposed approach with comparisons to the state of the art.
4) We also make our CVL-UMD data set and the source code open-source to facilitate future research.

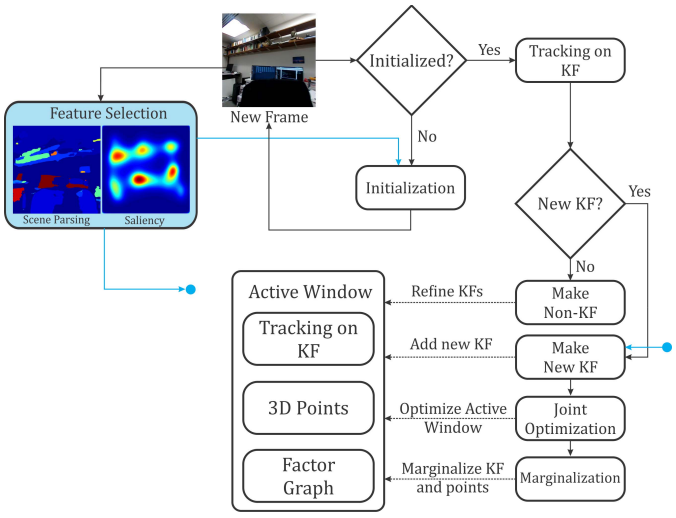

Fig. 2. Algorithmic overview of SalientDSO. Blue parts: our contributions. KF: keyframe.

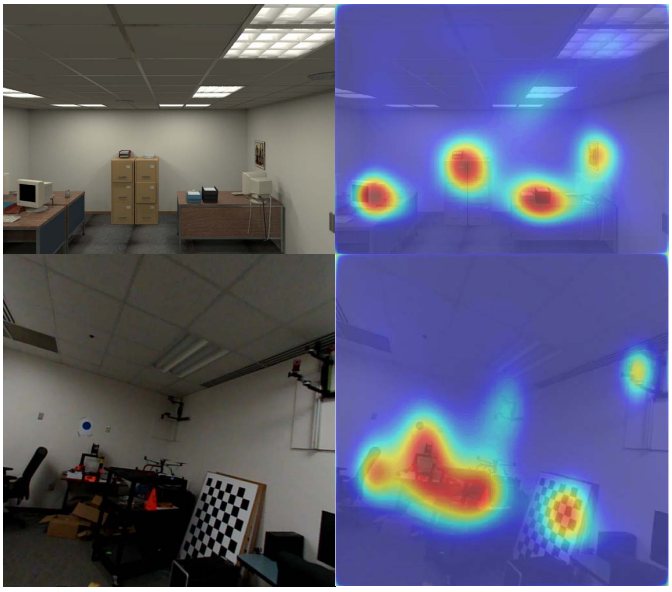

Fig. 3. Left column: input image. Right column: saliency overlaid on input image.

The rest of this paper is organized as follows. Section II presents the different parts of the proposed SalientDSO framework along with the preliminaries required. Section III describes the visual saliency and scene-parsing-driven point selection algorithm used in SalientDSO. Detailed experiments along with quantitative and qualitative results are given in Section IV. We finally conclude this paper in Section V with parting thoughts on future work.

II. SalientDSO Framework

SalientDSO's framework is composed of a preprocessing step and a VO backbone. The VO backbone is responsible for initializing and tracking camera pose and optimizing all model parameters. The preprocessing step involves the saliency prediction and scene parsing using deep convolutional neural networks and later using these outputs to select features/points. Fig. 2 shows the algorithmic overview of SalientDSO, where blue parts of the figure show our contributions (which constitute the preprocessing step). Each component of SalientDSO is discussed briefly in the following.

We adopt DSO [10] as the backbone VO in SalientDSO. In brief, DSO [10] proposed a direct sparse model to jointly optimize all parameters (camera intrinsics, camera extrinsics, and inverse-depth values for feature points) and perform windowed bundle adjustment. It contains a front end and a back end detailed in the following.

A. Front End

The front-end part of algorithm handles tracking, keyframe creation, outlier rejection, parameters initialization, candidate point activation, and marginalization as described in [10]. The candidate point selection in DSO is replaced by our novel approach described in Section III-C.

B. Back End

The back end contains a factor graph which performs continuous windowed optimization using the approach given in [23]. It optimizes E_{photo} using Gaussian–Newton algorithm in a sliding window manner. The error functions are defined as in [10].

III. Point Selection Based on Visual Saliency and Scene Parsing

A. Visual Saliency Prediction

Visual saliency is defined as the amount of attention a human would give to each pixel in an image. This is quantitatively measured as the average time a person's gaze rests on each pixel in the image. Prediction of saliency is a hard problem and data-driven approaches have lately excelled at this task. We adopt SalGAN [24] for saliency prediction in SalientDSO. Some sample results are shown in Fig. 3. One can clearly notice that walls, floors, and ceilings have lower probability of being fixated on, which is the main idea of the proposed framework.

B. Filtering Saliency Using Semantic Information

The saliency produced by SalGAN is concentrated around a fixation point inside the object and is fuzzy. The saliency map is robust with respect to the fixations remaining inside the object but the same fixation point is generally not obtained as the viewpoint and illumination changes [25], [26]. Note in Fig. 4 how the fixation point location is not robust but the protosegmentation obtained by the fixation is robust (the fixation point remains inside the same object) with respect to changing illumination and viewpoint conditions. In this section, we utilize semantic information to filter the saliency so as to use features from protosegmentation of an object which is more robust to viewpoint and illumination changes. The idea is to weigh down the saliency of uninformative regions, such as walls, ceilings, and floors. These regions are uninformative from the computer vision sense of them being weak corners. Semantically, objects/regions with unique

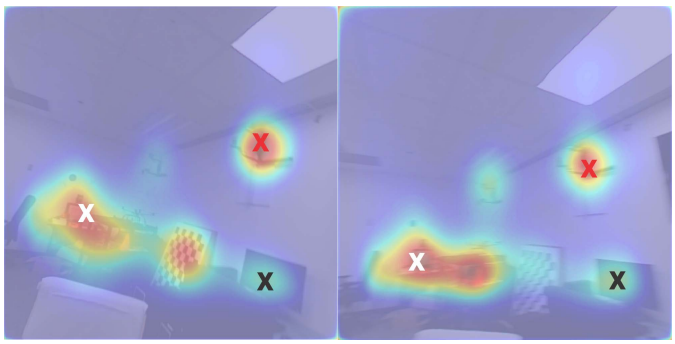

Fig. 4. Variation of saliency map due to changes in illumination and viewpoint. Note that the fixation remains inside the object but the saliency map varies. The crosses of respective color highlight the fixation in the respective images.

color or size or shape are more informative than regions such as walls, ceilings, and floors which have nonunique and nondiscriminative features.

To obtain semantic information from a scene, we adopt pyramid scene parsing (PSP) [27] for retrieving the semantic labels of every pixel in an image.

The predicted saliency map $\hat{S}$ is filtered using the PSPNet's per-pixel semantic output S

$$\hat{S}_j^{\text{weighted}} = w_C(C_j)\hat{S}_j. \tag{1}$$

Here, w_C represents the predefined weights obtained empirically for different classes. To smoothen and maintain a consistent saliency map for each class, each pixel is replaced by the median of saliency for its respective class

$$\hat{S}_j^{\text{final}} = \text{median}\{\hat{S}_i^{\text{weighted}}, \quad \forall i \in C_j\}. \tag{2}$$

All steps to generate $\hat{S}^{\text{final}}$ are summarized in Algorithm 1.

Algorithm 1 Saliency Prediction and Filtering

Data: Input image I, Predefined weights w_C
Result: Predicted final saliency $\hat{S}^{\text{final}}$

$\hat{S}$ = SalGAN(I);
C = PSPNet(I);
for $\forall \{x_j, y_j\} \in I$ **do**
  $\hat{S}_j^{\text{weighted}} = w_C(C_j)\hat{S}_j$;
end
for $\forall \{x_j, y_j\} \in I$ **do**
  $\hat{S}_j^{\text{final}} = \text{median}\{\hat{S}_i^{\text{weighted}}, \forall i \in C_j\}$;
end

C. Features/Points Selection

Instead of uniformly selecting candidate points from an image as in DSO, we select points based on saliency. This is very helpful where the scene has a lot of objects or clutter which can be found generally in indoor scenes.

First, we split an image into $K \times K$ patches. For a patch M_i, we not only compute the median of gradient as a region-adaptive threshold but also compute the median of saliency as a region-adaptive sampling weight sw_i. Therefore, for each patch, the sampling weight sw_i is computed as

$$sw_i = \text{median}\{\hat{S}_j^{\text{final}}, \quad \forall j \in M_i\} + s_{\text{smooth}} \tag{3}$$

where s_{smooth} is a Laplacian-smoothing term used to control the bias on a salient region and the probability of a patch M_i being sampled is

$$P_S(M_i) = \frac{sw_i}{\sum_{m \in M} sw_m}. \tag{4}$$

Second, once a patch M_i has been selected, we further split M_i into $d \times d$ blocks. For each block, we select the pixel with the highest gradient only if it surpasses the region-adaptive threshold. With this strategy, we can select points which are well distributed in this salient region. In order to extract information from where no high-gradient pixels are present, we follow the same approach as DSO and run two more passes to select pixels with a weaker gradient in a larger subregion with a lower gradient threshold and an increased d. A summary of the whole selection method is given in Algorithm 2.

Algorithm 2 Saliency-Based Points Selection

Data: Desired number of points N_{des}, s_{smooth}, $\hat{S}^{\text{final}}$
Result: Selected points

Initialize selected point set as $\{\emptyset\}$, $N_{\text{sel}} = 0$;
while $N_{sel} < N_{des}$ **do**
  Randomly select a patch M from distribution P_S;
  Split M into $d \times d$ blocks;
  for *each* $4d \times 4d$ *block* **do**
    for *each* $2d \times 2d$ *block* **do**
      for *each* $d \times d$ *block* **do**
        Select a point with the highest gradient which surpass the gradient threshold;
      end
      if *no selected point in this block* **then**
        Select a point with the highest gradient which surpass the weaker gradient threshold;
      end
    end
    if *no selected point in this block* **then**
      Select a point with the highest gradient which surpass the much weaker gradient threshold;
    end
  end
  $N_{\text{sel}} = N_{\text{sel}}$ + the number of selected points;
end

Fig. 5 shows the selected points for some example scenes. We compare our selection based on saliency to the uniform selection adopted by DSO. One can easily note that textureless and mostly identical parts, such as walls, floors, and ceilings, are down-weighted in our pipeline. As demonstrated in Section IV, this helps us trade the weak features on the floors and ceilings for weak features on objects where the saliency is generally higher—thus, in turn, making the feature selection more robust and object-centric.

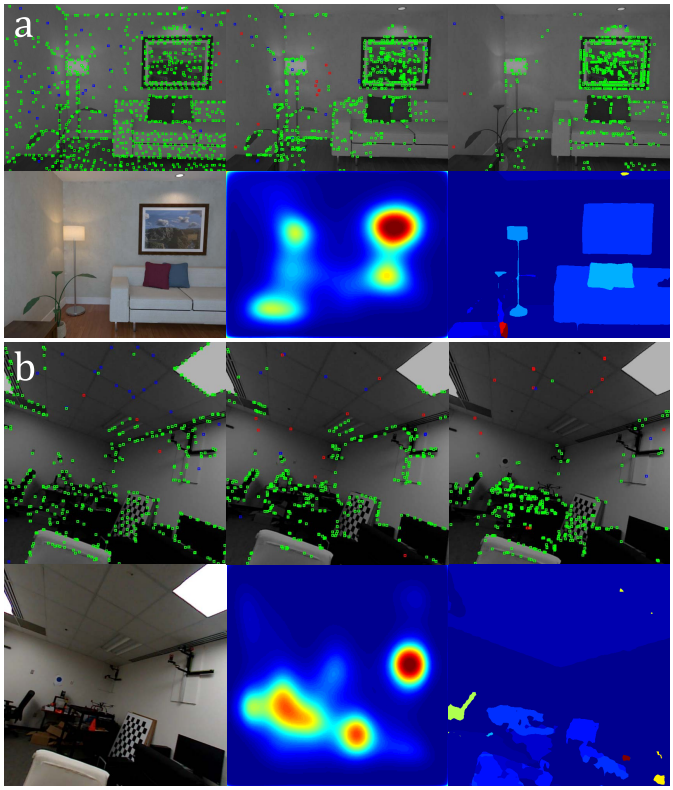

Fig. 5. Point selection using different schemes. (a) and (b) Images from ICL-NUIM and CVL-UMD data sets, respectively. Top rows: Features selected using DSO's scheme, saliency only, saliency+scene parsing (left to right). Bottom rows: input image, saliency, and scene-parsing output (left to right). Notice how using saliency + scene parsing removed all noninformative features.

IV. Experimental Results and Analysis

In this section, we comprehensively evaluate SalientDSO on various data sets.

1) *ICL-NUIM Data Set [28]*: This data set provides two scenes and four different trajectories for each scene which are obtained by running continuously on real image data and finally used in a synthetic framework for obtaining ground-truth.
2) *TUM monoVO Data Set [29]*: This data set provides 50 sequences comprising over 100-min videos. It ranges from indoor corridors to wide outdoor scenes. In our experiments, we only evaluate all methods on indoor sequences {sequence_$(1 - -18, 26, 28, 35 - -38, 40)$}. Only the indoor sequences are chosen because the usage of saliency obtained by human gaze is meaningful for indoor cluttered scenes.
3) *CVL-UMD Data Set*: We collected this data set and is now available at prg.cs.umd.edu/SalientDSO.html. The data were collected using a Parrot SLAMDunk [30] sensor suite. The data from the left camera are used in the experiments.

Different parameters used for running the experiments are shown in Table II. For ICL-NUIM data set, photometric correction is not required. To comprehensively evaluate the proposed method, we run each sequence in both forward and backward direction ten times.

TABLE II
Parameter Settings for Different Data Sets

	TUM	ICL-NUIM	CVL-UMD
Num of active keyframes N_f	7	7	7
Num of active points N_p	2000	2000	1200
Global gradient constant g_{th}	7	3	7
Patch size K	8	8	8
Photometric correction	Yes	Not required	Not available

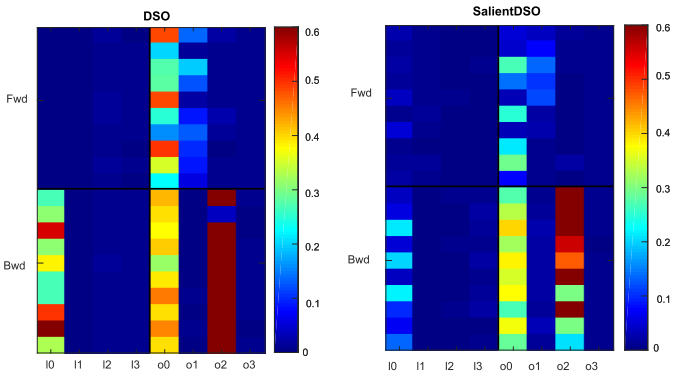

Fig. 6. Comparison of evaluation results for ICL-NUIM data set. Left: DSO. Right: SalientDSO. Each square corresponds to a color-coded error. Note that SalientDSO almost always has lower error than its DSO counterpart.

Also, note that the neural networks employed in this paper are adapted directly from their respective papers [24], [27] and were *not fine-tuned* as the concept of scene parsing and saliency are generic. The SalGAN [24] was trained on Salicon data set [31]. The Salicon data set collects the saliency data as a probability of visual attention by aggregating mouse trajectories and contains 10 000 training images, 5000 validation images, and 500 testing images. The PSPNet [27] was trained on the ADE20K data set [32]. The ADE20K data set contains annotations of 150 object classes all annotated by a single annotator for consistency. It contains 20 210 training images, 2000 validation images, and 3000 testing images.

A. Quantitative Evaluation

Fig. 6 shows the absolute trajectory root-mean-square error (RMSE_{ate}) on ICL-NUIM data set (each rectangle shows a different run). Using visual saliency-driven features, SalientDSO performs better in accuracy as compared to DSO. We also report alignment error e_{align} on TUM monoVO data set in Fig. 7. We disable the semantic filtering when we evaluate the proposed method on the TUM monoVO data set, since this data set provides only grayscale images and outputs from PSPNet are inaccurate and noisy for grayscale images. In Tables III and IV, we compare our method to DSO and ORB-SLAM [5] on the ICL-NUIM and the TUM monoVO data sets. For a detailed description of the error metrics used in Tables III and IV, see [33] and [29], respectively. DSO and ORB-SLAM [5] are the current state-of-the-art direct and feature-based monocular VO methods. The results for DSO and ORB-SLAM are taken from [10]. ORB-SLAM is

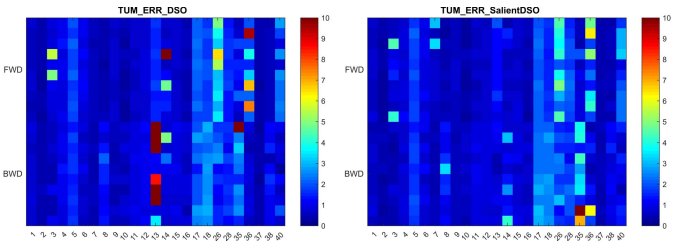

Fig. 7. Comparison of evaluation results for TUM data set. Left: DSO. Right: SalientDSO. Note that SalientDSO almost always has lower error than its DSO counterpart. Note that, for the TUM data set, the scene parsing was turned off as TUM data set only provides grayscale images and scene-parsing outputs are very noisy for grayscale images.

TABLE III

RMSE_{ATE} ON ICL-NUIM DATA SET IN m

	Forward			Backward		
Sequence	ORB	DSO	SalientDSO	ORB	DSO	SalientDSO
ICL_l0	0.01	**0.003**	0.022	**0.01**	-	0.112
ICL_l1	0.02	**0.004**	0.009	0.04	0.003	**0.003**
ICL_l2	0.06	0.012	**0.004**	0.19	0.010	**0.005**
ICL_l3	0.03	0.006	**0.004**	0.05	**0.008**	0.013
ICL_o0	0.21	0.320	**0.140**	0.41	0.399	**0.336**
ICL_o1	0.83	0.094	**0.055**	0.68	**0.006**	0.020
ICL_o2	0.37	0.012	**0.008**	0.32	0.582	**0.512**
ICL_o3	0.65	**0.007**	0.009	0.06	**0.006**	0.008
Overall Avg.	0.271	0.057	**0.031**	0.218	0.144*	**0.126**

* indicates average taken only on sequences which completed.

TABLE IV

e_{ALIGN} ON TUM MONOVO DATA SET IN m

	Forward			Backward		
Sequence	ORB	DSO	SalientDSO	ORB	DSO	SalientDSO
seq_01	3.02	**0.59**	0.60	1.73	0.72	**0.60**
seq_02	16.12	0.36	**0.33**	3.23	**0.43**	0.44
seq_03	3.42	1.75	**1.55**	1.42	0.59	**0.50**
seq_04	9.95	0.98	**0.82**	5.95	1.00	**0.76**
seq_05	-	1.86	**1.77**	-	**1.55**	1.66
seq_06	-	0.97	**0.93**	1.25	**0.73**	0.81
seq_07	1.69	**0.55**	1.14	2.02	**0.44**	0.48
seq_08	436.00	**0.36**	0.44	2.63	**1.28**	1.47
seq_09	2.04	0.65	**0.58**	0.67	**0.52**	0.53
seq_10	2.52	0.35	**0.34**	1.43	0.61	**0.61**
seq_11	7.20	0.62	**0.58**	2.99	**0.87**	0.89
seq_12	2.98	0.75	**0.67**	3.10	1.01	**0.84**
seq_13	5.13	1.54	**1.27**	2.59	8.96	**0.81**
seq_14	13.27	2.89	**0.71**	2.10	**1.35**	1.69
seq_15	2.90	0.71	**0.71**	1.90	0.88	**0.81**
seq_16	2.40	0.47	**0.45**	1.58	0.72	**0.67**
seq_17	12.29	2.10	**2.10**	**1.50**	2.13	2.50
seq_18	14.64	1.77	**1.52**	-	2.62	**2.47**
seq_26	28.46	3.98	**3.60**	4.62	**1.66**	1.89
seq_28	19.17	**1.48**	1.88	3.57	**1.47**	1.65
seq_35	14.09	1.10	**0.84**	16.81	**5.48**	9.97
seq_36	1.81	4.01	**3.25**	1.69	**0.70**	1.46
seq_37	0.60	**0.35**	0.40	1.30	**0.37**	0.46
seq_38	-	0.55	**0.50**	24.77	1.10	**1.03**
seq_40	-	**2.04**	2.16	18.93	**0.87**	1.04
Overall Avg.	28.55*	1.31	**1.17**	4.69*	1.52	**1.44**

* indicates average taken only on sequences which completed.

TABLE V

COMPARISON OF SUCCESS RATE BETWEEN DSO AND SALIENTDSO ON CVL-UMD DATA SET

Sequence	DSO	SalientDSO
CVL_01_Fwd	53%	**65%**
CVL_01_Bwd	59%	**92%**
CVL_02_Fwd	73%	**96%**
CVL_02_Bwd	71%	**91%**

a full-fledged SLAM framework with loop closure and global alignment, whereas DSO and SalientDSO are merely odometry frameworks. To make the comparison fair, loop-closure detection and relocalization have been turned off for ORM-SLAM. The missing values in these tables represent tracking failures. We achieve similar or better performance on most sequences. The improvement is not significant on the TUM monoVO data set because most of the sequences involve a traversal through a hallway where there are no local salient objects or features for saliency prediction to work well. This makes SalientDSO's performance close to that of traditional DSO.

The claim in this paper is that the usage of visual saliency should result in more robust features than just using image-gradient-based features as in DSO. The intuition behind this claim is that visual saliency includes high-level semantics which inherently makes the features more robust. To support this claim, we anticipate that SalientDSO should perform much better than DSO when the number of points is very low (as low as 40 points). To demonstrate this claim, we evaluate each CVL-UMD sequence. We run each sequence in both forward and backward direction 100 times, with an extremely low point density of $N_p = 40$. The results are shown in Table V. We define failure as either an optimization failure or tracking loss. Our proposed method is much more robust and predicts an accurate trajectory, whereas DSO has a much higher failure rate and its trajectory and projected point cloud shows significant drift in scale and position. An example of trajectory and projected point cloud is shown in Fig. 8. This experiment highlights the robustness of features chosen in SalientDSO for cluttered indoor scenes and how this will be useful for robots with very low computation power due to the less computational and memory requirements when N_p is low.

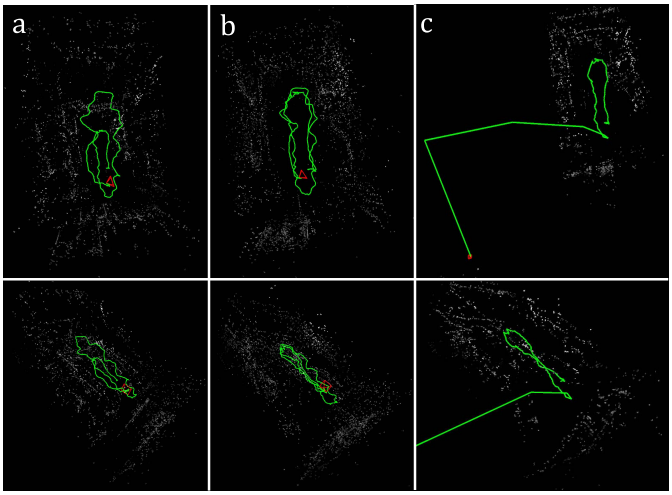

Fig. 8. Comparison of outputs for $N_p = 40$ (very few features). (a) Success case of DSO with a large amount of drift. (b) Success case of SalientDSO. (c) Failure case of DSO where the optimization diverges due to very few features. Note that SalientDSO can perform very well in these extreme conditions showing the robustness of the features chosen.

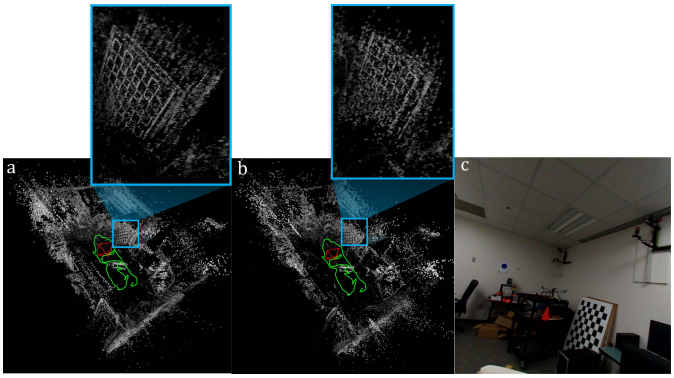

Fig. 9. Comparison of drift. (a) DSO's output. (b) SalientDSO's output. (c) Image corresponding to crop shown in the inset. Observe that SalientDSO's output has the checkerboard from different times more closely aligned as compared to DSO. Here, $N_p = 1000$.

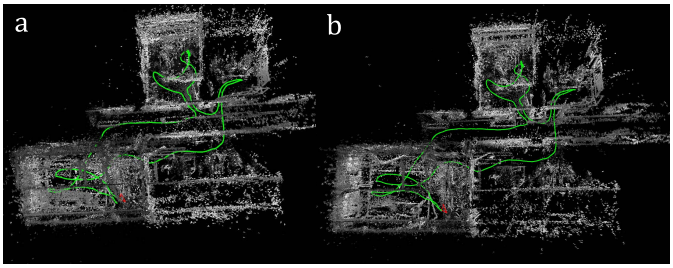

Fig. 10. Sample outputs for TUM sequence_1. (a) DSO. (b) SalientDSO. Here, $N_p = 1000$.

B. Qualitative Evaluation

Examples of the reconstructed scenes of sequences CVL_01 and TUM sequence_01 are shown in Figs. 9 and 10, respectively. Although both reconstructed scenes look similar, one could observe that the amount of drift in SalientDSO is much less compared to DSO (refer to the zoomed part of Fig. 9), that is, the checkerboard's features from different loops align better in our approach—ideally, the features from different loops should align perfectly. However, due to drift, the checkerboard appears as two different planes from two different loops; the spread between the planes of the checkerboard indicate drift. Note that the spread between the planes of the checkerboard is much higher in DSO as compared to SalientDSO. Instead of sampling random high gradient points, sampling salient points that are considered to be more informative by the visual saliency model improves the robustness of VO.

V. Conclusion

We introduce the philosophy of attention and fixation to VO. Based on this philosophy, we develop SalientDSO that brings the concept of attention and fixation based on visual saliency into VO to achieve robust feature selection. We provide thorough quantitative and qualitative evaluations on the ICL-NUIM and the TUM monoVO data sets to demonstrate that using salient features improves the robustness and accuracy. We also collect and publicly release a new CVL-UMD data set with cluttered scenes for mapping. We show the robustness of our features by very low drift VO with as low as 40 features per frame. Our method takes about a second per keyframe (*not every frame*) for the computation of saliency and scene parsing on an NVIDIA Titan-Xp GPU and the remaining computations run real time at 30 fps on an Intel Core i7-6850K 3.6-GHz CPU. In the near future, we plan to extend our method to outdoor environment. We also consider to implement our method on hardware to make the complete pipeline real time. Finally, we also make our source code open-source for enabling future research.

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[3] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2320–2327.

[4] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system, *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[6] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *Proc. Joint Pattern Recognit. Symp.*, 2010, pp. 11–20.

[7] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Oct. 2015, pp. 298–304.

[8] S. A. Scherer and A. Zell, "Efficient onbard RGBD-SLAM for autonomous MAVs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Nov. 2013, pp. 1062–1068.

[9] N. J. Sanket, C. D. Singh, K. Ganguly, C. Fermüller, and Y. Aloimonos, "GapFlyt: Active vision based minimalist structure-less gap detection for quadrotor flight," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 2799–2806, Oct. 2018.

[10] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[11] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Jun. 2017, pp. 1722–1729.

[12] X. Yu, C. Fermüller, C. L. Teo, Y. Yang, and Y. Aloimonos, "Active scene recognition with vision and language," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 810–817.

[13] A. S. Ogale, C. Fermuller, and Y. Aloimonos, "Motion segmentation using occlusions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 988–992, Jun. 2005.

[14] C. Fermüller and Y. Aloimonos, "Tracking facilitates 3D motion estimation," *Biol. Cybern.*, vol. 67, no. 3, pp. 259–268, 1992.

[15] A. Mishra, Y. Aloimonos, and C. L. Fah, "Active segmentation with fixation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 468–475.

[16] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 333–356, 1988.

[17] J. Bohg *et al.*, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1273–1291, Dec. 2017.

[18] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Auton. Robot.*, vol. 22, no. 2, pp. 177–196, 2018.

[19] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. M. M. Montiel, "Towards semantic SLAM using a monocular camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Sep. 2011, pp. 1277–1284.

[20] L. An, X. Zhang, H. Gao, and Y. Liu, "Semantic segmentation–aided visual odometry for urban autonomous driving," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 5, Oct. 2017, Art. no. 1729881417735667.

[21] T. Dang, C. Papachristos, and K. Alexis, "Visual saliency-aware receding horizon autonomous exploration with application to aerial robotics," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2018, pp. 2526–2533.

[22] H. Liu, F. Sun, and X. Zhang, "Robotic material perception using active multi-modal fusion," *IEEE Trans. Ind. Electron.*, to be published. doi: 10.1109/TIE.2018.2878157.

[23] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[24] J. Pan *et al.* (2017). "SalGAN: Visual saliency prediction with generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1701.01081

[25] O. L. Meur, "Robustness and repeatability of saliency models subjected to visual degradations," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 3285–3288.

[26] C. Kim and P. Milanfar, "Finding saliency in noisy images," *Proc. SPIE*, vol. 8296, 2012. doi: 10.1117/12.905760.

[27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[28] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Hong Kong, China, May/Jun. 2014.

[29] J. Engel, V. Usenko, and D. Cremers. (2016). "A photometrically calibrated benchmark for monocular visual odometry." [Online]. Available: https://arxiv.org/abs/1607.02555

[30] (2018). *Parrot SLAMDunk*. [Online]. Available: http://developer.parrot.com/docs/slamdunk/

[31] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "SALICON: Saliency in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1072–1080.

[32] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5122–5130.

[33] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. A. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2012, pp. 573–580.

Authors' photographs and biographies not available at the time of publication.