Application-oriented Design Space Exploration for SLAM Algorithms

Sajad Saeedi†, Luigi Nardi†, Edward Johns†, Bruno Bodin⋆, Paul H. J. Kelly†, and Andrew J. Davison†

***Abstract*—In visual SLAM, there are many software and hardware parameters, such as algorithmic thresholds and GPU frequency, that need to be tuned; however, this tuning should also take into account the structure and motion of the camera. In this paper, we determine the complexity of the structure and motion with a few parameters calculated using information theory. Depending on this complexity and the desired performance metrics, suitable parameters are explored and determined.**

Additionally, based on the proposed structure and motion parameters, several applications are presented, including a novel active SLAM approach which guides the camera in such a way that the SLAM algorithm achieves the desired performance metrics. Real-world and simulated experimental results demonstrate the effectiveness of the proposed design space and its applications.

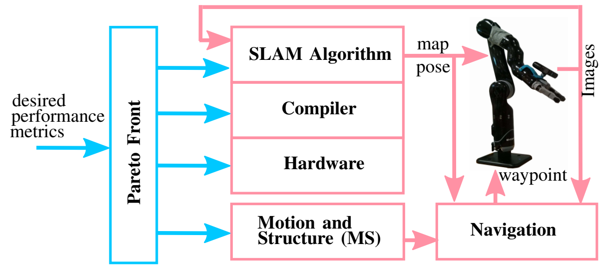

Fig. 1: A SLAM-based application is composed of four design spaces: SLAM algorithm, compiler, hardware, and motion and structre (MS), where the latter is a new concept introduced in this paper. Each space has a set of parameters (blue arrows) chosen from a Pareto front, depending on the required performance metrics. The MS parameteres are calculated using information theory. These parameters together with images are used to control the motion of the camera, such that the desired performance metrics are maintained.

I. Introduction

Recently within the Simultaneous Localization and Mapping (SLAM) and robot vision community, it has been a controversial issue whether SLAM is solved or not. To answer this question, we need to consider three main factors as defined by Cadena et al. in [1]: *robot*, *environment*, and *performance*. In other words, the answer depends on the *robot* (its motion, resources, batteries, sensors, ...), the *environment* (indoor, outdoor, dynamic, ...), and the required *performance* (the desired accuracy, success rate, latency, ...). For instance, 2D grid-based SLAM in indoor environments with a required reconstruction error of below 0.01 m could be considered solved. Similarly, visual SLAM is also considered almost solved, but in some applications, when the robot has very fast dynamics or the environment is highly dynamic, the performance of the mapping and localization degrades. Therefore, research on SLAM is entering a new era where robust performance and application-oriented SLAM is the focus.

There are several different discrete paradigms for SLAM algorithms, including sparse [2], semi-dense [3], dense [4], and semantic [5]. At the next level, there are possible major choices between components of these algorithms (e.g. type of feature, type of surface representation, etc), and finally, parameter choices within a particular algorithm. The choice of the algorithm is dependent on the application, the available resources, and the required performance metrics. There have been measures and benchmarks for SLAM systems for several years now, and these have been widely used to compare and tune the performance of different algorithms and systems. The majority of these have concentrated on accuracy; mainly of trajectory, because that is straightforward to independently measure, but sometimes of mapping accuracy too.

However, the performance of a SLAM algorithm on an accuracy benchmark actually tells us little about how useful it would be for a particular application. SLAMBench [6] showed how the usefulness of benchmarks could be broadened in an important dimension by considering efficiency of performance on different computing platforms. A SLAM algorithm which is useful for a high accuracy industrial mapping application is almost certainly not the right choice for a low power embedded platform like a drone. This has started to open up research on Design Space Exploration (DSE) [7], where a high level search is made through the possible operating parameters of a SLAM system, in order to find the combinations which work best in terms of an appropriate compromise between accuracy and efficiency. In general, the results of DSE are represented by a Pareto front of possible operating points, where each point on the front represents an optimum set of parameters given the desired performance metrics. But still, the scene and motion are fixed in SLAMBench; all variations of algorithms are tested on a certain synthetic scene dataset with a certain camera motion.

In reality, different applications need to work in different environments; and have varying specifications with regard to motion. If a drone must use visual SLAM to navigate through a forest, it will be flying fast past complex, nearby trees; while a robot vacuum cleaner navigates rather slowly on a ground plane, but must deal with a scene which is often distant and textureless. How can we perform design space exploration for SLAM systems as a whole, taking into account this range of applications with different constraints and requirements? It would seem that the specific qualities of the motion and structure involved in an application would need many parameters to specify — the typical linear and rotational velocities and accelerations of motion; the structure complexity of the scene; the average scene depth; the level of texture, and so on.

The hypothesis of this paper is that we can use a small set of parameters as a very useful proxy for a full description of the setting and motion of a SLAM application. We call these Motion and Structure (MS) parameters, and define them based on information theory. Specifying and searching through MS parameters in design space exploration allows us to focus within the wide range of possible operating points represented by an accuracy/efficiency Pareto front. Using the MS parameters, we are able to identify how challenging the

†Robot Vision Group, Imperial College London, UK
⋆School of Informatics, University of Edinburgh, UK

environment is with a given camera motion, and thus choose a set of more suitable hardware and software parameters from the Pareto front. One of the applications of the proposed MS parameters, as shown in this paper, is the active SLAM with robotic platforms (Fig. 1). Unlike other information theoretic methods, such those which try to maximize mutual information or information gain [8] for rapid exploration, in our method we propose to limit the information divergence, to ensure that the SLAM system is robust with respect to the structure of the observed scene.

A. Contributions

The contributions of this work are as follows:

- Introducing a comprehensive design space, including motion and structure space, for real-time applications,
- Parameterising the motion and structure space with information theory, and
- Proposing several applications based on the MS parameters, including an active SLAM algorithm.

The rest of the paper is organised as follows: Section II presents background and literature review. In Section III the proposed motion and structure space is introduced. In Section IV the design space exploration is explained. In Section V, several applications of the proposed motion and structure space are presented. In Section VI, experiments are presented, and in Section VII, conclusions and future works are presented.

II. Background and Literature Review

In this section, three topics are presented: performance metrics, design space exploration, and information theory.

A. Performance Metrics

SLAM algorithms are compared based on various performance metrics such as accuracy, robustness, processing cost, and etc [9], [10], [11], [12]. Strum et al. improve trajectory metrics, absolute trajectory error (ATE) and relative pose error (RPE), by evaluating the root mean squared error over all time indices of the translational components [11].

Other important metrics are related to the quality of the map, such as reconstruction completeness (RCM), defined as the reconstructed percentage of the ground truth points [13], and reconstruction error (RER), defined as the error between the reconstructed and the ground truth map. As an example, in ElasticFusion [4], where the map is shown by surfels, RER is determined by running iterative closest point (ICP) algorithm on the point cloud models of the world and the map. The error of the point cloud matching is used as RER.

Execution time (EXT), memory usage (MEM), and energy consumption (ENE) per frame are other important metrics which are usually taken into account in real-world applications and on mobile devices.

B. Design Space Exploration

The design parameters of a SLAM algorithm are categorised as either software parameters, including algorithmic and compiler parameters, or hardware parameters.

Algorithmic parameters are algorithm dependent. For instance, in KinectFusion [14], the ICP convergence threshold, volume resolution, and pyramid level iterations are such algorithmic parameters. *Compiler parameters* operate at the compiler level and affect the way that the hardware executes the algorithm. Vectorisation and compiler flags for the precision of mathematical operations are examples of such parameters. *Hardware parameters* include the number of active CPU cores and the GPU processor frequency. By proper selection and tuning of these parameters, the objective is to achieve the desired performance metrics; however, the augmented hardware and software variables form a large vector that is not easy to tune manually. Additionally, there are multiple choices for the desired parameters which are shown by a Pareto front. Figure 7 demonstrates the Pareto front highlighted in green, where each point on the Pareto front is an optimal answer. For every non-Pareto point, there is a point on the front which is better in at least one metric. A user can choose the desired point from the front depending on the trade-off between metrics.

In the recent paper of SLAMBench [6], the idea of adopting the KinectFusion algorithm to run on four different platforms with default algorithmic parameters was proposed. SLAMBench uses ICL-NUIM dataset [15] to do experiment.

Bodin et al. proposed the idea of design space exploration (DSE) which tries to optimise the hardware and software parameters to achieve some of the desired performance metrics, including ATE, ENE, and EXT [7]. The methodology of their work is based on quantifying these indices by playing the KinectFusion algorithm using the ICL-NUIM dataset on two different platforms and exploring the design space parameters.

Zia et al. apply a similar concept in [16], but at a small scale, to only algorithmic parameters of the KinectFusion and LSD-SLAM [3] algorithms. They have done their experiments using ICL-NUIM and TUM-RGBD [11] datasets.

C. Information Divergence

Information theory and its concepts such as entropy and mutual information has many applications in robotics and perception, including path planing [17], SLAM [18], and exploration [19]. In this paper, information divergence is used to assess the quality of mapping and localization.

In information theory, information divergence, which is a measure of difference between two probability distribution functions, has been used in many different fields such as image processing, speech processing, and machine learning [20].

As an information divergence measure, the Kullback-Leibler divergence also called KL divergence or relative entropy, is a natural distance measure that uses Shannon's entropy. For a discrete random variable with dimension d, such as$X = (X_1, \ldots, X_d) \in \mathbb{R}^d$ with a probability distribution function of $p(x_1, \ldots, x_d)$, the entropy is defined as:

$$H(X) = \sum_{x_1,\ldots,x_d} p(x_1, \ldots, x_d) log \frac{1}{p(x_1, \ldots, x_d)}. \quad (1)$$

If the random variables $X_i, i = 1, \ldots, d$ are independent, equation (1) becomes:

$$H(X) = \sum_{i=1,\ldots,d} H(X_i). \quad (2)$$

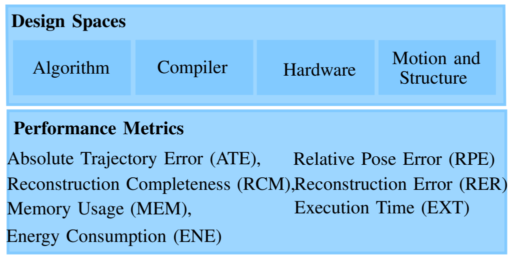

Fig. 2: The SLAM design space includes four main domains: 1) algorithm, 2) compiler, 3) hardware, and 4) motion and structure. The choice of the parameters for these domains are evaluated using various performance indices such as absolute trajectory error (ATE), reconstruction error (RER), or execution time (EXT).

If X_is are independent and identically distributed, $H(X)$ is

$$H(X) = dH(X_i). \tag{3}$$

Entropy $dH(X_i)$ is the upper bound for the entropy that can be achieved. In other words, the upper bound for $H(X)$ is when X_is are independent and identically distributed. Similarly, by extending the definition in equation (1), the relative entropy or KL divergence distance for two distributions, $p(X)$ and $q(X)$, is defined as:

$$\delta(p||q) = \sum_{x_1,\ldots,x_d} p(x_1,\ldots,x_d) log \frac{p(x_1,\ldots,x_d)}{q(x_1,\ldots,x_d)}. \tag{4}$$

When $p(\cdot)$ and $q(\cdot)$ are equal, the distance is zero.

III. Motion and Structure Parameter Space

This section explains the Motion and Structure (MS) design space. If a camera mounted on a quadrotor experiences a sudden change in the view of the scene due to the fast dynamics of the quadrotor, depending on the depth of the scene, the SLAM algorithm may fail or succeed to process the following frames because tracking is difficult when sequential images have very different appearances. Therefore it is important to quantify the limits of the physical motions in different environments. In other words, it is desired to represent this complex dependency of motion and structure with a minimum number of parameters which are also easy to compute. For sparse SLAM, Civera et al. achieved this goal by decomposing the state space into metric parameters and dimensionless parameters [21]. The dimensionless parameters are used to tune the SLAM filter without any assumption about the scene. The structure of the scene also offers important cues as to which parameters to use, and we later address this.

One way to take into account the behaviour of the motion in a structure is to refer to the sensory data. The information gained, from one frame to another, tells us about the motion of the camera relative to the environment. As it is shown, there is a correlation between the change of the information from one frame to the next, and the desired performance metrics. Extremely high rates of change will result in failure of SLAM, as expected. The MS design space identifies the maximum change permitted for a SLAM algorithm to achieve the desired performance metrics.

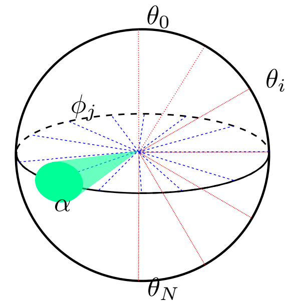

Fig. 3: Surface of a unit sphere is divided into equal patches to bin the normal vectors of a depth image. First N equal inclination angles are created (θ_i, shown in red dotted lines). For each inclination, M equal azimuth angles are created (ϕ_j, shown in blue dotted lines). A unit vector, identified by θ_i and ϕ_j, demonstrates a bin which attracts any normal that falls inside an influence region, shown by α.

A. Divergence of Sensor Information

In the rest of the work, it is assumed that the sensor operates in a realistic environment, i.e. the sensor is not blind, and the structure has minimum texture to be mapped. If images are modelled by probability distributions, by knowing the magnitude of divergence in information from one distribution to another, we are able to determine the motion of the sensor in an environment. In an extreme case, a zero divergence means there is no motion. A large divergence may indicate that either the sensor is moving very fast, or the environment has rapidly varying structure.

1) **Approximate KL Divergence for Intensity Images:** We treat an image as a very high-dimensional discrete random variable. An approximate probabilistic model can then be generated by assuming that pixels are individually independent random variables. The reason that this is an approximate model is that in practice the pixels are correlated through the geometry of the environment; however, modelling the geometry is not a trivial task. In this work, for intensity images, an approximate probability distribution model is generated by making a normalised histogram of the intensities of the pixels. This is similar to the model that Shannon created to model English words [22]. The key is that the normalised histogram is an estimate of the underlying probability of each pixel's intensity.

For two intensity images, I_t and I_{t-1}, the normalised histogram of intensity values is considered as their distribution functions. Typically for intensity images, $P = 256$ bins are considered, where each bin is associated with an integer $u = 0, ..., 255$. If the distributions of the images are indicated by $\mathcal{I}_t$ and $\mathcal{I}_{t-1}$, the intensity information divergence is:

$$\delta_I(t) \triangleq KL(\mathcal{I}_t||\mathcal{I}_{t-1}) = \sum_{l=0}^{P} \mathcal{I}_t(u) \log \frac{\mathcal{I}_t(u)}{\mathcal{I}_{t-1}(u)}, \tag{5}$$

where subscript I indicates that the distribution and the divergence distance are derived from the intensity images, and δ_I is the KL divergence. $\mathcal{I}(u)$ is the u^{th} bin of distribution $\mathcal{I}$.

2) **Approximate KL Divergence for Depth Images:** To create depth distributions, depth values could also be binned similarly; however, this is not trivial given the unbounded range of these depth values. Instead, for two consecutive depth images, D_t and D_{t-1}, their probability distribution functions are defined using normal vectors of the depth points. To generate the distributions, the method used in [23] has been adopted. First, from the depth images, for each point, a normal vector is calculated. Then the normal vectors are binned to created a histogram. To bin the normal vectors, a unit sphere is partitioned into equal patches (see Fig. 3). Patches are identified by their central azimuth and inclination angles. To have equally distributed patches, regularly-spaced inclination angles are selected by dividing the inclination range (i e. $[0,\, \pi]$) to N equally-distributed angles (equation (6)). For each inclination, the azimuth angles are selected by dividing the azimuth range (i e. $[0,\, 2\pi]$) to M equal angles (equation (7)). Note that as we get closer to the poles, M decreases. The $\lfloor\cdot\rfloor$ sign denotes the integer part operation.

$$\theta_i = \pi\frac{i}{N}, \quad i = 1..N \tag{6}$$

$$\phi_j(\theta_i) = 2\pi\frac{j}{M}, \quad j = 1..M, \quad M = \lfloor 2N\sin\theta_i\rfloor + 1 \tag{7}$$

Once the bins are created, the k^{th} normal vector n_k, $k = 1..L$, contributes to bin i, j based on the angle between n_k and v_{ij}, where v_{ij} represents bin i, j in Cartesian coordinates:

$$w_{ij}^k = \begin{cases} 0 & \text{if} \cos^{-1}(n_k.v_{ij}) > \alpha \\ \dfrac{n_k.v_{ij} - \cos\alpha}{1-\cos\alpha} & \text{else} \end{cases} \tag{8}$$

In equation (8), α is the angular range of influence for each bin. Based on these weights, the spherical distribution is:

$$\mathcal{D}(i,j) = \sum_{k=1}^{L} w_{i,j}^k, \qquad i = 1..N,\ j = 1..M \tag{9}$$

After calculating the contribution of all normals to the bins, the histogram is normalized to sum to one. For two distributions, $\mathcal{D}_t$ and $\mathcal{D}_{t-1}$, the depth information divergence, δ_D, is:

$$\delta_D(t) \triangleq KL(\mathcal{D}_t||\mathcal{D}_{t-1}) = \sum_{i,j}\mathcal{D}_t(i,j)\log\frac{\mathcal{D}_t(i,j)}{\mathcal{D}_{t-1}(i,j)}, \tag{10}$$

B. Motion and Structure Design Space

There is a direct relationship between the KL divergence distance of intensity and depth images, and the performance metrics. A larger divergence means more outliers during image alignment, which introduces more error. If we want to perform better outlier rejection by relying on more iterations or more accurate algorithms, like RANSAC, the hardware requirements increase. In general, the relationship between the divergence and the metrics is not easily proved analytically, and thus it has been shown experimentally here.

To efficiently represent the MS design space with information divergence, for a trajectory of T frames $(1..T)$, the maximum information divergences for intensity and depth images, $\mathcal{M}_\mathcal{I}$ and $\mathcal{M}_\mathcal{D}$, are introduced:

$$\mathcal{M}_I = \max\left(\delta_I(t)\big|_{t=1:T}\right) \tag{11}$$

$$\mathcal{M}_D = \max\left(\delta_D(t)\big|_{t=1:T}\right) \tag{12}$$

To demonstrate the relationship between these divergence values and performance metrics, a dataset is tested using ElasticFusion [4]. To generate data streams with larger information divergence, frames were skipped in regular intervals. Then the absolute trajectory error (ATE) is calculated. Fig. 4 demonstrates the absolute trajectory error versus divergence for the ICL-NIUM dataset (stream lr kt1 and lr kt2). This figure shows that higher information divergence corresponds to higher trajectory error.

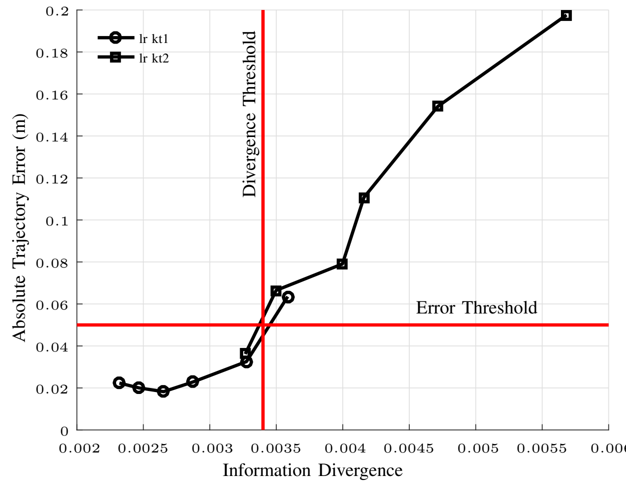

Fig. 4: Absolute trajectory error versus information divergence for two streams of ICL-NUIM dataset (lr kt1, lr kt2). As information divergence increases, the overall ATE increases. This leads us to the idea that by limiting the information divergence actively, i.e. through the motion feedback, the desired ATE can be achieved.

These maximum divergence values parameterise the motion and structure simultaneously. In other words, for a desired ATE, the motion in a given structure should be such that the frame to frame divergence should not exceed these parameters.

IV. Design Space exploration

In this section, the design space exploration with the four design spaces, shown in Fig. 2, is explained. The design space exploration is performed with ElasticFusion [4] on an Intel machine. For simplicity, the experiments are performed only on algorithmic and MS parameters. To evaluate the design parameters, ATE and EXT performance metrics are calculated.

A. Design Parameters

The ElasticFusion algorithm is parameterised by the following parameters. For a detailed description, please refer to original paper by Whelan et al. [4].

- Depth cutoff: Cutoff distance for depth processing. Range: $[0-10]$ m, default: 3 m.
- ICP / RGB tracking weight: This weight determines the ratio of ICP to RGB tracking in visual odometry. Range: $[0-1]$, default: 0.1.
- Confidence threshold: Surfel confidence threshold. Range: $[0-12]$, default: 10.

For the motion and structure parameters, maximum intensity information divergence, and maximum depth information divergence, are used. These two parameters were introduced in Equations (11) and (12). To determine these parameters, a dataset sequence was played with frames being dropped at different rates, and the maximum information divergence was calculated across that sequence. Dropping frames actually occurs in real-world applications, i.e. when there is limited buffer or processing resource, the unprocessed frames are simply discarded. While the parameters for the algorithmic, hardware, and compiler domains were generated in advance, the parameters for the MS space is produced on the fly.

B. Procedure

We wish to determine the Pareto front for those parameters which are defined above. To generate one point on the Pareto plane, we first generate a random sample of the algorithmic space parameters. We then specify a frame drop rate, and by running the algorithm with these parameters on the corresponding image sequence, the EXT and ATE metrics are calculated, together with the corresponding MS parameters (maximum information divergence). This process continues, each time adding a Pareto point, until we have the Pareto front determined. The Pareto front is later used to specify design space parameters based on the trade-off between different performance metrics.

V. Applications of Design Space Exploration

In this section, four different scenarios are presented which show how the proposed MS parameters and design space exploration are used in real-world applications to meet the objectives of a mission or limitation of the resources. These scenarios are active frame management, run-time adaptation, dataset difficulty level assessment, and active SLAM. Of these, the active SLAM algorithm is explained in detail and some experimental results are presented in the next section.

A. Active Frame Management

In real-world applications, optimising resources such as battery is very important. One of the applications of the design space exploration is the ability to decide when to process a frame. If two consecutive frames are statistically very similar, by processing them, we are able to gain more confidence in the map and the pose of the camera; however, this is at the cost of spending other important resources such as battery. In this situation, it is desirable to simply drop the second frame to save the battery. Obviously when there are unlimited resources, it is desirable to process all frames. To manage frames actively, for each frame its information divergence with respect to the previous frame is calculated. If the divergence is less than a threshold, the frame is not passed to the SLAM pipeline. The threshold can be dynamic and could be a function of the available resources such as battery or the processing resources.

B. Run-time Adaptation

Assume that after the design space exploration, a set of parameters have been identified from the Pareto front; these guarantee acceptable performance metrics according to a defined maximum information divergence. If for any reason, the information divergence is higher than the expected values, there is risk of having poor performance. However, this can be counteracted by choosing another set of parameters, which may require higher allocation and consumption of the available resources, but can deal with the higher divergence. In other words, using the proposed method, it is possible to have multiple sets of parameters, and in extreme situations, our method can easily switch from one set of parameters to another to guarantee the required performance metrics.

Dataset	No	Max	Mean	Variance
ICL-NUIM	lr kt0	0.0250	0.0026	0.0014
	lr kt1	0.0183	0.0026	0.0012
	lr kt2	0.0427	0.0032	0.0023
	lr kt3	0.0352	0.0032	0.0023

TABLE I: Difficulty level metrics using information divergence.

C. Dataset Difficulty Level Assessment

When proposing a new SLAM algorithm, a de facto is to compare the results with other algorithms by testing them on known datasets. So far there is no measure to assess the difficulty level of the datasets, and thus, the comparison with datasets may not be able to reveal all strengths or weaknesses of a new SLAM algorithm. As a standard metric, the proposed information divergence, without considering software and hardware parameters, can easily be used to assess the difficulty of different datasets. This can be achieved by assigning statistics of the information divergence, such as mean and variance, to the sequence of the data in each dataset. Additionally, the divergence metrics can be evaluated per unit of time. Table I shows some of these statistics for ICL-NUIM datasets (only intensity divergence for simplicity). According to [4], in ICL-NUIM, datasets lr kt2 and lr kt3 are more difficult than lr kt0 and lr kt1 based on the reported performance metrics. These difficult trajectories have a higher difficulty score.

D. Active SLAM with Information Divergence

Active SLAM, also known as active vision, view path planning (VPP), or next best-view (NBV), is the problem of determining the optimal camera motion (in some sense) to perform mapping [24]. Active SLAM is closely related to the exploration problem [25], where the objective is to map an unknown environment completely.

There are several works that perform active SLAM with sensors such as lasers for 2D/3D mapping [26], [27], but Davison and Murray were the first who integrated motion with stereo visual SLAM [28] where their objective was to minimize the trajectory error. Most active SLAM algorithms are either based on maximizing mutual information [8], [29], or maximizing information gain [30], [31] for increased coverage, decreased pose uncertainty, or dense mapping purposes. But our objective is to maintain robustness and achieve the desired performance metrics by controlling the incoming information flow. In other words, we guide the camera such that the information divergence is not more than the permitted maximum information divergence defined in equations (11) and (12).

1) **Active SLAM based on Information Divergence:** Fig. 5 shows the block diagram of the system. The *SLAM* block implements the ElasticFusion algorithm [4]. The resulting pose and map are used in the *motion planning* block. $\mathcal{M}_I$ and $\mathcal{M}_D$ are the MS parameters that are used for motion planning. The most recent images, I_{t-1} and D_{t-1}, together with the predicted next images, generated from the current map, are used to determine the next best waypoint for the *controller* block. The controller guides the robot using inverse kinematics. The details of this controller are beyond the scope of the paper.

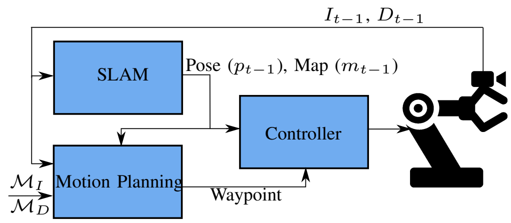

Fig. 5: Active SLAM with a robotic arm: image and depth frames along with Pareto fronts are used to determine the best next waypoint to control the amount of the information that is sent to the SLAM pipeline.

Algorithm 1 explains the proposed motion planning in detail. Inputs to the algorithm are the previous intensity and depth images, $(I_{t-1},\, D_{t-1})$, the previous pose and map estimates, $p_{t-1},\, m_{t-1}$, and the maximum allowed intensity and depth divergence parameters, $(\mathcal{M}_I, \mathcal{M}_D)$. Based on these inputs, the algorithm determines the best rotation and translation, T, to maintain the information divergence below the threshold.

In line 1, Δ_I and Δ_D, which contain divergence values for candidate poses, are initialised. In line 2, the space around the current pose is decomposed to reachable rotation and translation motions. The decomposed space includes seven translations along the axes of the current local frame: no translation, up, down, left, right, forward, and backward. For each translation, there are seven rotations in the local frame including no rotation, roll right, roll left, pitch forward, pitch backward, yaw anti-clockwise, and yaw clockwise. $\mathcal{T}$ contains the set of rotations and translations for the decomposed space. With this simple decomposition, there are 49 elements in $\mathcal{T}$. In line 4, the candidate global poses of the camera, given the previous pose and the next potential pose transformations, are calculated. In line 5, for each of the candidate poses, depth and intensity images are predicted by projecting the current map, m_{t-1}, on the camera plane. $(\hat{I}^i_t, \hat{D}^i_t)$ are predicted intensity and depth images for the i^{th} candidate pose $\hat{p}^i_t$. In lines 6 and 7, for each of the predicted images, the divergence with respect to the last intensity and depth images are calculated. $\Delta_I(i)$ and $\Delta_D(i)$ contain the corresponding divergences for the i^{th} candidate pose. Given the predicted images and their divergences, in line 9, a pair of predicted intensity and depth images are chosen which has the optimum divergence distance to the divergence parameters. In this line, two thresholds is introduced, defined as a percentage of the maximum allowed intensity and depth information divergence, denoted by ρ_I and ρ_D. Note that these two thresholds control the exploratory behavior of the algorithm. If these parameters are zero, the algorithm wants to keep the camera almost stationary, and if they are set to 1, the algorithm wants to move the camera to the locations where the image will provide maximum allowed information, defined by $\mathcal{M}_I$ and $\mathcal{M}_D$. Also, λ in this line is a weight parameter, used to adjust the significance of depth over intensity in the optimisation. Since the criterion has a finite number of elements, i.e. only 49 different candidate poses, the optimization is performed exhaustively. Finally, in line 10, the rotation and translation commands associated with the chosen intensity and depth images, are selected and passed to the controller.

The proposed motion planning is a local algorithm and does not provide a global destination for the camera. To provide global planning, in line 9, by adding more constraints, the optimisation for the next motion can be combined with any globally planned trajectory. This allows us to guide the camera with global optimality, whilst providing acceptable information divergence.

Algorithm 1 Active SLAM using information divergence parameters

Require: Last intensity and depth frames : $(I_{t-1},\, D_{t-1})$,
Last pose and map: $p_{t-1},\, m_{t-1}$,
Motion and structure parameters: $(\mathcal{M}_I, \mathcal{M}_D)$
Ensure: Next best move : T

1: $\Delta_I \leftarrow \{\}$, $\Delta_D \leftarrow \{\}$
2: $\mathcal{T} \leftarrow \text{decompose}(p_{t-1})$
3: **for** $i = 0, \quad i < |\mathcal{T}|\}$ **do**
4: $\quad \hat{p}^i_t \leftarrow p_{t-1} \oplus \mathcal{T}(i)$
5: $\quad (\hat{I}^i_t, \hat{D}^i_t) \leftarrow \text{project}(\hat{p}^i_t, m_{t-1})$
6: $\quad \Delta_I(i) \leftarrow \text{divergence}(\hat{I}^i_t, I_{t-1})$
7: $\quad \Delta_D(i) \leftarrow \text{divergence}(\hat{D}^i_t, D_{t-1})$
8: **end for**
9: $i^\star \leftarrow \text{argmin}_i\{|\Delta_I(i) - \rho_I \mathcal{M}_I)| + \lambda|\Delta_D(i) - \rho_D \mathcal{M}_D|\}$
10: $T \leftarrow \mathcal{T}(i^\star)$
11: return T

VI. Experiments

In this section, we evaluate how our method can optimise parameters to achieve certain desired metrics. Then we provide in-depth exploration of the application to active SLAM, and present both simulated and real-world experiments with a camera mounted on a robotic arm.

A. Design Space Exploration

This experiment demonstrates the usefulness of DSE in providing better performance metrics using information divergence. Fig. 6 shows maximum ATE vs. EXT per frame for various divergence values in the ICL-NUIM dataset (stream lr tr0). In the legend, the highlighted marks have been sorted from the highest divergence ($\times$) to the lowest ($\circ$). In Fig. 6, as divergence increases, ATE and EXT increase.

Next, for one of the divergence values, DSE is implemented as explained in Section IV-B to find the suitable algorithmic parameters. For the point marked with $\diamond$, maximum ATE is 2 cm, and EXT is approximately 0.038 s per frame. In Fig 7, this point has been shown by a black diamond as default parametric configuration. All other points show the results of DSE. The Pareto front has been shown by a green curve. Using DSE, the ATE for this divergence can be reduced down to 1 cm and EXT can be reduced to less than 0.02 s.

B. Active SLAM in Simulation

This experiment demonstrates the concept of performing active SLAM, in which the motion of the camera is controlled to adjust the information flow to the SLAM pipeline. In the simulation, a pair of intensity and depth images are rendered from a known world model (ICL-NUIM living room) given

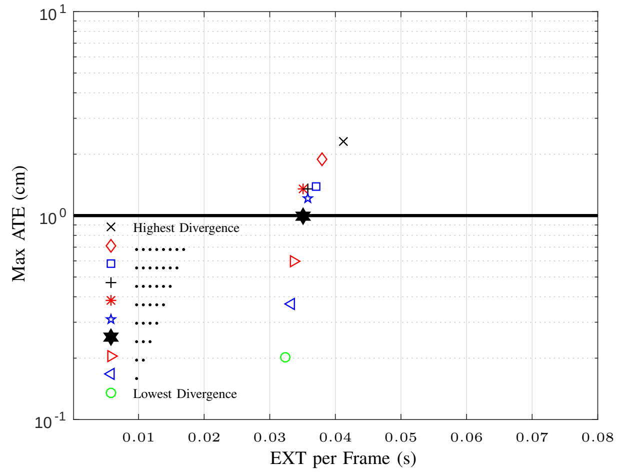

Fig. 6: ATE versus EXT for different divergence values, from the highest divergence (marked by ×) to the lowest divergence (marked by ○). The point corresponding to ◇ has ATE of 1 cm. For this point, DSE will be performed in Fig. 7.

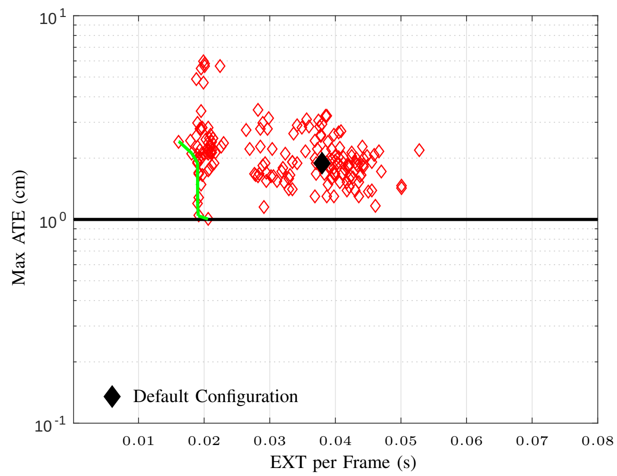

Fig. 7: For the points corresponding to ◇ in Fig. 6, DSE is performed and the results are shown. From the Pareto front (shown in green), the EXT from 0.038 s can be reduced to 0.018 s, or the ATE can be reduced form 2 cm to 1 cm.

the current pose of the camera. These images are processed by SLAM, and also by the the motion planner to decide what the next pose of the camera should be. Once the next pose is known, the camera is guided to the desired pose, and the process of rendering images, SLAM, and motion planning continues recursively. To render images from the 3D model, Persistence Of Vision Raytracer, POVRay [1], is used. POVRay renders much more realistic images compared to similar tools such as Gazebo [2]. In the simulation, two different motion planning algorithms are tested: random walk and the proposed active SLAM. In the random walk, for each frame, one transformation is chosen from the 49 different transformations available (combination of 7 translations and 7 rotations as explained in Section V-D), while in the active SLAM, a transformation that optimises the information divergence is chosen (Algorithm 1). Fig. 8 shows a demonstration of 49 different intensity image predictions and their divergence scores (depth images are not shown for the sake of brevity). The experiment was repeated twice (Table II). In the *free motion* experiment, rotation and translation were changing as explained. In the

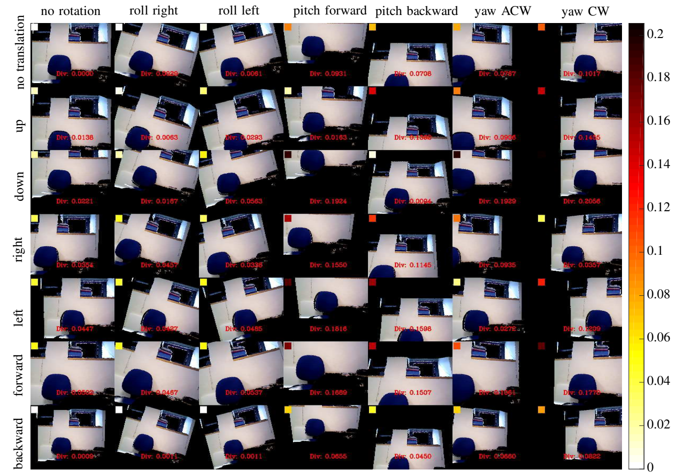

Fig. 8: Predicted images for active SLAM with their divergence distance with respect to the current image. Divergence values are also depicted with color-coded patches on top-left corner of each predicted image. The color bar (right) shows the divergence values.

Experiment	Algorithm	ATE	RER
Free Motion	Random Walk	0.2343	0.0773
	Active SLAM	**0.1549**	**0.0518**
Fixed Translation	Random Walk	0.0854	0.0809
	Active SLAM	**0.0582**	**0.0726**

TABLE II: Performance of random walk versus active SLAM in simulation.

fixed translation experiment, the camera was translating along a straight line, and the rotation was optimised (or randomly selected). Table II compares the performance metrics for these experiments. The results show that the active SLAM generated better results in terms of performance metrics.

C. Active SLAM with Robotic Arm

This experiment demonstrates the active SLAM algorithm with a robotic arm. Fig 1 shows the Kinova Mico Arm [3] used for active SLAM. An ASUS RGB-D camera was mounted on the arm, and as with the previous experiment, random walk and active SLAM (Algorithm 1) are compared.

The experiments were done in four different environments, labelled as *window*, *table*, *wall*, and *carpet*. In each environment, each algorithm was run 10 times. Repeated experiments serve as a measure of the robustness of the algorithm in dealing with uncertainties rising from minor changes in illumination, or inaccuracies of the response of the controller or actuator to the commands.

For the random walks, different initial seeds were used everytime. Due to the lack of ground truth information from the real environments, the consistency of the generated map was evaluated manually as either a success or failure of SLAM. If duplicates of one object were present in the map, it was considered as failure. The generated maps are available for inspection [4]. Fig. 9 shows these results. As the figure demonstrates, in all four cases, active SLAM performs better than random walk. Particular performance difference is noted in the *carpet* experiment, where random walk failed in all 10 tries, and active SLAM succeeded in five out of ten tries by moving in and out and maintaining smaller information divergence than random walk.

[1] http://www.povray.org/
[2] http://gazebosim.org/
[3] http://www.kinovarobotics.com/
[4] https://imperialcollegelondon.box.com/s/aemavn2o0vo57obe3bhot62ww522ksnc

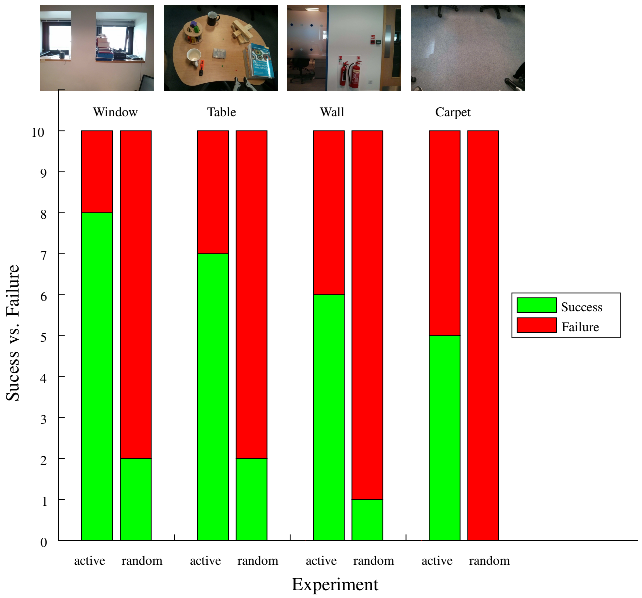

Fig. 9: Success vs. failure rate when mapping the same environment with different motion planning algorithms: active SLAM and random walk.

VII. Conclusion and Future Work

This paper introduced a new domain for the design space exploration of the SLAM problem, called Motion and Structure (MS) space. The new domain is represented by parameters, calculated using information divergence, that can be used to meet the desired performance metrics. An active SLAM algorithm was also developed based on the MS parameters, and we showed how our method can be used to guide camera motion optimally to ensure robust performance. We also presented a design space exploration experiment which demonstrated that suitable MS parameters can be incorporated with other design space parameters, to yield a Pareto front.

In future work, we propose to use the information divergence metric to evaluate several other real-world robotic applications, including run-time adaptation. Another direction to explore is adding global path planning constraints to the active SLAM algorithm, to enable autonomous navigation as well as ensuring robust performance. Additionally, we are exploring improvements to the divergence measure, such as introducing spatial windowing across the image for histogram generation, and using the Earth mover's distance to provide tolerance to small illumination changes.

Acknowledgment

This research is supported by Engineering and Physical Sciences Research Council (EPSRC), grant no. EP/K008730/1, A Panoramic View of the Many-core Landscape (PAMELA).

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, "Simultaneous localization and mapping: Present, future, and the robust-perception age," 2016.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29, no. 6, pp. 1052–1067, 2007.

[3] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *ECCV*, 2014, pp. 834–849.

[4] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *RSS*, 2015.

[5] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *CVPR*, 2013, pp. 1352–1359.

[6] L. Nardi, B. Bodin, M. Z. Zia, J. Mawer, A. Nisbet, P. H. J. Kelly, A. J. Davison, M. Luján, M. F. P. O'Boyle, G. Riley, N. Topham, and S. Furber, "Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM," in *ICRA*, May 2015.

[7] B. Bodin, L. Nardi, Z. Zeeshan, H. Wagstaff, G. S. Shenoy, M. Emani, J. Mawer, C. Kotselidis, A. Nisbet, M. Lujan, B. Franke, P. H. J. Kelly, and M. O'Boyle, "Integrating algorithmic parameters into benchmarking and design space exploration in dense 3D scene understanding," in *Parallel Architectures and Compilation Techniques*, 2016.

[8] T. Vidal-Calleja, A. J. Davison, J. Andrade-Cetto, and D. W. Murray, "Active control for single camera SLAM," in *ICRA*, 2006.

[9] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kummerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J. Tardos, "A comparison of SLAM algorithms based on a graph of relations," in *IROS*, 2009, pp. 2089–2095.

[10] R. Kummerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of SLAM algorithms," *Autonomous Robots*, vol. 27, no. 4, pp. 387–407, 2009.

[11] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IROS*.

[12] S. Schwertfeger and A. Birk, "Evaluation of map quality by matching and scoring high-level, topological map structures," in *ICRA*, 2013, pp. 2221–2226.

[13] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: Probabilistic, monocular dense reconstruction in real time," in *ICRA*, 2014, pp. 2609–2616.

[14] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *ISMAR*, 2011, pp. 127–136.

[15] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *ICRA*, 2014.

[16] M. Z. Zia, L. Nardi, A. Jack, E. Vespa, B. Bodin, P. H. J. Kelly, and A. J. Davison, "Comparative Design Space Exploration of Dense and Semi-Dense SLAM," in *ICRA*, 2015.

[17] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-theoretic planning with trajectory optimization for dense 3D mapping," in *RSS*, 2015.

[18] A. J. Davison, "Active search for real-time vision," in *Proceedings of the International Conference on Computer Vision (ICCV)*.

[19] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *RSS*, 2005.

[20] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

[21] J. Civera, A. J. Davison, and J. M. Montiel, "Dimensionless monocular SLAM," in *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis*, 2007, pp. 412–419.

[22] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[23] T. Fiolka, J. Stückler, D. A. Klein, D. Schulz, and S. Behnke, *SURE: Surface Entropy for Distinctive 3D Features*. Springer Berlin Heidelberg, 2012, pp. 74–93.

[24] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *IJRR*, 2011.

[25] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *ICRA*, 2005, pp. 661–666.

[26] R. Valencia, J. V. Mir, G. Dissanayake, and J. Andrade-Cetto, "Active pose SLAM," in *IROS*, 2012, pp. 1885–1891.

[27] S. Kriegel, T. Bodenmller, M. Suppa, and G. Hirzinger, "A surface-based next-best-view approach for automated 3D model completion of unknown objects," in *ICRA*, 2011, pp. 4869–4874.

[28] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 7, pp. 865–880, 2002.

[29] M. Bryson and S. Sukkarieh, "Observability analysis and active control for airborne slam," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 1, pp. 261–280, 2008.

[30] A. Kim and R. M. Eustice, "Active visual SLAM for robotic area coverage: Theory and experiment," *The International Journal of Robotics Research*, 2014.

[31] C. Forster, M. Pizzoli, and D. Scaramuzza, "Appearance-based active, monocular, dense reconstruction for micro aerial vehicles," in *RSS*, 2014.