From Feature Detection in Truncated Signed Distance Fields to Sparse Stable Scene Graphs

Daniel R. Canelhas, Todor Stoyanov, Achim J. Lilienthal
Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden

***Abstract*— With the increased availability of GPUs and multi-core CPUs, volumetric map representations are an increasingly viable option for robotic applications. A particularly important representation is the truncated signed distance field (TSDF) that is at the core of recent advances in dense 3D mapping. However, there is relatively little literature exploring the characteristics of 3D feature detection in volumetric representations. In this paper we evaluate the performance of features extracted directly from a 3D TSDF representation. We compare the repeatability of Integral invariant features, specifically designed for volumetric images, to the 3D extensions of Harris and Shi & Tomasi corners. We also study the impact of different methods for obtaining gradients for their computation. We motivate our study with an example application for building sparse stable scene graphs, and present an efficient GPU-parallel algorithm to obtain the graphs, made possible by the combination of TSDF and 3D feature points. Our findings show that while the 3D extensions of 2D corner-detection perform as expected, integral invariants have shortcomings when applied to discrete TSDFs. We conclude with a discussion of the cause for these points of failure that sheds light on possible mitigation strategies.**

I. Introduction

Since 2011, the advancement of GPU technology coupled with the commercial availability of affordable depth-sensing video cameras has sparked an interest in dense 3D mapping in real time. One of the main scientific and technological achievements at the start of this trend is undoubtedly *Kinect Fusion*, by Newcombe et al. [1], shortly followed by several extensions [2][3] and alternative formulations of the original problem and solution [4][5]. At the core of these algorithms is an elegant method for volumetric integration of depth information into a truncated signed distance field (TSDF).

A TSDF computed as a weighted sum of signed distances, measured along the rays of a perspective camera, is shown to represent the maximum likelihood estimate for the surface corresponding to a set of depth images, as its zero-level isosurface [6]. TSDFs thus offer a map representation that implicitly represents the mean estimate of the surface location and its variance. Given a TSDF, novel viewpoints can be easily synthesized by casting rays into the volume using e.g. sphere-tracing [7]. The depth maps obtained in this way tend to be of higher quality, and produce better results when used for 2.5D feature detection and feature descriptor matching [8]. In this work we are interested in investigating the stability of feature detection directly in the 3D TSDF, instead of on the depth-maps sampled from it or polygonal meshes extracted from it through a marching cubes algorithm. Finding salient regions such as edges and corners in 2D is a problem that has been studied thoroughly, but the research on volumetric counterparts has not been given the same attention. In a relatively recent work by Yu et al. [9] six volumetric adaptations of 2D feature detectors are comparatively evaluated on 3D volumes derived from point-sets by Gaussian kernel density estimation. Although TSDFs are different from density images, our choice of corner detectors is partly motivated by their evaluations on MRI scans.

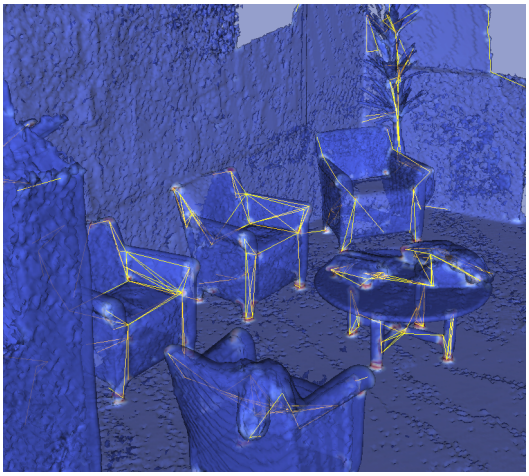


Fig. 1. Lounge dataset reconstructed as a TSDF, visualized as its triangulated zero-set, and its associated SSSG

In this study, we apply the volumetric adaptations of Harris [10] and Shi-Tomasi [11] corner detectors and compare the effects of different choices of gradient estimators on their stability. We evaluate these corner detectors against two types of integral invariants [12] specifically designed for volumetric image domains and signed distance fields, but whose performance on TSDFs (let alone TSDFs generated from actual sensor data) is currently unknown. We test all of the above for stability with respect to rigid-body transformations. While the properties of the volumetric extensions of standard 2D features such as Harris corners are well understood at present, they serve to set the results of the integral invariants into perspective, and allow us to understand the trade-off between their relative computational complexity and performance in a more meaningful way. In this article we do not concentrate on evaluating saliency detectors for polygonal meshes (e.g. [13], [14], [15]), as such evaluations have already been reported in literature (see [16]). Instead, our main focus here lies in evaluating native saliency detectors that operate directly on SDF models and are directly applicable to online usage scenarios.

Lastly, to showcase the unique applications that the combination of TSDFs and feature detection in the 3D space enable, we present an algorithm to efficiently extract a novel graph structure called Sparse Stable Scene Graphs (SSSG) that summarizes the main characteristics of a scene as a graph of geometrically linked salient features as illustrated in Fig. 1. We ilustrate the utility of the SSSG by means of a proof-of-concept RANSAC based place matching application. To summarize, our main contributions in this work are:

- a stability analysis of the volumetric extensions of Harris and Shi & Tomasi corners with respect to the choice of derivative estimation strategy,
- a thorough analysis of the applicability of Integral invariant features in TSDFs,
- a novel GPU-parallel method for building sparse stable scene graphs (SSSG) from TSDFs, given a set of feature points.

II. Feature Detection

Feature detection and description are typical steps in many object recognition tasks and localization steps of SLAM algorithms. Focusing on salient features avoids computation on indistinct regions that are likely to provide little useful information in subsequent descriptor matching steps. Of critical importance for the success of matching descriptors is that the process that selected where they should be computed is repeatable, thus our focus will be primarily on the stability of feature detectors with regards to perturbations of the voxel grid. In II-A we will define TSDFs in more detail. In II-B and II-C we describe the Harris and Shi & Tomasi corner detectors and their applications to 3D images. Because the aforementioned features are gradient-based, we dedicate some space in II-D to discuss the rationale behind different choices of gradient estimators and explain their derivation. In II-E we review the concepts of integral invariant features.

A. Truncated Signed Distance Field (TSDF)

A distance field is an implicit surface representation that encodes the location of an arbitrary surface Φ by providing, for a given query point $\boldsymbol{x} \in \mathbb{R}^3$, the signed distance to the closest surface point on Φ. The sign indicates if $\boldsymbol{x}$ is inside (negative) the volume bounded by Φ or outside (positive). The surface itself is thereby encoded as the zero-crossing of the signed distance field.

$$dist(\boldsymbol{p}, \Phi) : \mathbb{R}^3 \rightarrow \mathbb{R} \tag{1}$$

Since the environment is not fully observable from any given viewpoint it is not possible to construct and maintain a full SDF (see Fig. **??**) reliably from depth maps. However, a TSDF, which limits distances to be bounded by a range of $[d_{min} \;\; d_{max}]$, can be constructed in real-time [17] using incremental, local updates with approximate signed distances, measured along the lines of sight of the sensor. Given a sufficient number of observations and an appropriate weighing scheme for combining them, the projective distances tend to approximate the closest distance metric with good accuracy, as illustrated in Fig 2.

B. Harris Corners

Most feature detection methods apply a response function over the entire image domain and retain the locations for which the function both exceeds a threshold and is also locally maximal. One such response function is the minimum sum of squared differences (SSD) [18] within a region around a candidate location. This can be interpreted as giving a high score to points where the image derivative is not small in any given direction. Harris [10] approximates the Hessian of the SSD as

$$\boldsymbol{H}_2 = \frac{1}{|w|} \sum_w \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2}$$

where w represents a square (or round, if desired) window around the candidate point, $|w|$ is the number of elements in w, for normalization and I_x, I_y are the estimated gradients of the image in horizontal and vertical directions, respectively. The extension to 3D is straightforward.

$$\boldsymbol{H}_3 = \frac{1}{|w|} \sum_w \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{bmatrix} \tag{3}$$

The response R for a given voxel is computed as:

$$R = det(\boldsymbol{H}_3) - k \;\; Tr(\boldsymbol{H}_3)^3 \tag{4}$$

with k being an empirical constant for which a typical value (in the volumetric case) is 0.001 [19]. The above formulation is algebraically equivalent to the following, using eigenvalues.

$$R = \prod_{i=1}^{dim} (\lambda_i) - k \;\; \sum_{i=1}^{dim} (\lambda_i)^3 \tag{5}$$

C. Good Features to Track

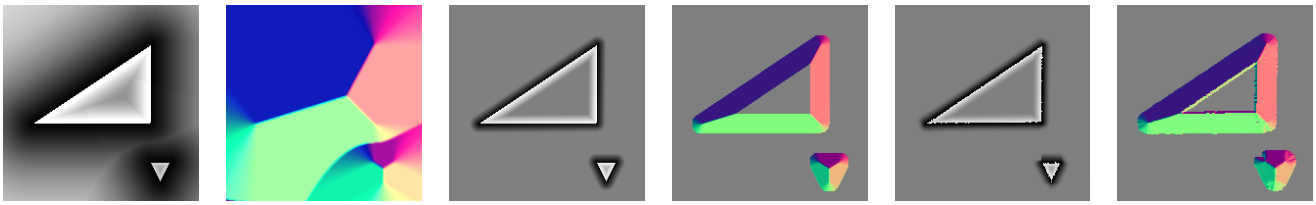
Shi and Tomasi [11] argued that when images undergo general affine transformations a better choice for R is simply

$$R = min(\lambda_1, \cdots \lambda_{dim}) \tag{6}$$

However, in our three dimensional image setting the affine warps typically associated with projective geometry are not likely to occur, thus we expect the assumption of pure rigid-body motion to be sufficient in most cases. Nonetheless, we test the use of Eq. (6), too.

D. Derivatives

A point to be made against response functions based on gradients is that gradients are susceptible to noise and that this in turn reduces the stability of the resulting features. Image derivatives may be obtained by a simple central differencing scheme but are often calculated by convolution with a filter kernel that represents the weighted average of several central difference computations. By including pixel samples from a neighbourhood around the point of interest, some robustness to noise is obtained at the expense of locality. The same applies to voxels. A common choice of filtering kernel in 2D is the 3x3 Sobel-Feldman operator [20] which can be interpreted as the application of a low pass filter (an integer



(a) (b) (c) (d) (e) (f)

Fig. 2. In (c) and (d) we see the true TSDF and its gradient-map, computed from the scene shown in Fig. **??**. In (e) and (f) we see the TSDF and gradients produced by reconstructing the same scene with measurements generated via a virtual moving depth-sensor.

approximation to the Gaussian kernel) and differentiation. See Eq. (7) for the example of the derivative filter in the horizontal direction where $*$ denotes a 2D convolution or equivalently, (8) using ordinary matrix multiplication.

$$SoFe_h \in \mathbb{R}^{3\times3} = \begin{bmatrix}1 & 0 & -1\end{bmatrix} * \begin{bmatrix}1\\2\\1\end{bmatrix} = \begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix} \quad (7)$$

$$SoFe_h \in \mathbb{R}^{3\times3} = \begin{bmatrix}1\\0\\-1\end{bmatrix} \begin{bmatrix}1 & 2 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & -1\\2 & 0 & -2\\1 & 0 & -1\end{bmatrix} \quad (8)$$

By convolving the differencing operator and blurring operator with themselves, i.e.,

$$\begin{bmatrix}1 & 0 & -1\end{bmatrix} * \begin{bmatrix}1 & 0 & -1\end{bmatrix} = \begin{bmatrix}1 & 2 & 0-2 & -1\end{bmatrix} \quad (9)$$

$$\begin{bmatrix}1 & 2 & 1\end{bmatrix} * \begin{bmatrix}1 & 2 & 1\end{bmatrix} = \begin{bmatrix}1 & 4 & 6 & 4 & 1\end{bmatrix} \quad (10)$$

one obtains filter coefficients that can be combined, in the same manner as in Eq. (8) to produce a 5×5 derivative kernel.

Since the Sobel-Feldman operator is an approximation to the derivative of the Gaussian function, the latter is worth some consideration as an option, too. For our analysis, we compute the analytic derivatives and directly form the 3D filter kernels [21] of size $3\times3\times3$ and $5\times5\times5$ with variances $\sigma_3 = 0.95$ and $\sigma_5 = 1.25$, respectively. Keeping in mind that our use for gradients is a means for obtaining repeatable feature points, we are led to also investigate another set of derivative kernels, optimized for rotation invariance, as proposed by Scharr [22]. Generating the volumetric filter kernels from the 1D coefficient vectors is analogous to the 2D case and is detailed in Algorithm 1. The variables $\boldsymbol{g}$, $\boldsymbol{b}$, $direction$, n are column vectors containing the derivative and blurring filter coefficients, derivative direction and kernel size (3 or 5), respectively. The derivative and filter coefficients appear in Table I , for reference.

Algorithm 1 Computing the volumetric filter kernels from their 1-D coefficient vectors

Require: $\boldsymbol{b}$, $\boldsymbol{g}$, $direction$, n

Allocate $n \times n \times n$ elements for $\boldsymbol{K}$
switch ($direction$)
case "x"**:**
  $\boldsymbol{S} \leftarrow \boldsymbol{b}\boldsymbol{g}^T$
  for all z in 1 to n **do**
    $\boldsymbol{K}_z \leftarrow \boldsymbol{s}_z\boldsymbol{f}^T$ { $\boldsymbol{K}_z$, is the z-th slice of $\boldsymbol{K}$ and $\boldsymbol{s}_z$ is the z-th column of $\boldsymbol{S}$}
case "y", "z"**:**
  Analagous, with $\boldsymbol{S} \leftarrow \boldsymbol{g}\boldsymbol{b}^T, \boldsymbol{b}\boldsymbol{b}^T$, $\boldsymbol{K}_z \leftarrow \boldsymbol{s}_z\boldsymbol{b}^T, \boldsymbol{s}_z\boldsymbol{g}^T$
Return $\boldsymbol{K}$

E. Integral Invariant Features

Integral invariants were first introduced by Manay et al. [23] and are local shape descriptors defined as integrals over a rotationally symmetric neighbourhood. The *local area invariant* and *distance invariant*, were both shown to provide a local and efficient estimate for *mean curvature* of a shape in 2D with robustness to noise. Pottmann et al. [12] presented integral invariants defined via three-dimensional signed distance fields and made an extension of the local area invariant to the volumetric case. Here we further extend the study of signed distance and volume invariants to their application on truncated signed distance fields. Both of these features have their domains defined as the volume bounded by a sphere, centred around a *surface* point $\boldsymbol{p}$. The assumption that computation is carried out centred on surface points implies that voxel-based methods are a poor fit, since the probability of a voxel being centred exactly on the surface i.e. the zero-level of the TSDF, is very small. However, Pottmann et al. [12] mathematically show that these features are stable to perturbations of the query point location, if the integration radius is sufficiently large. This reported stability encourages our attempt to apply integral invariants even in the discrete case.

The volume invariant $V_r(\boldsymbol{p})$ is the integral of the indicator function $1_D(\boldsymbol{x})$ which returns 1 if $\boldsymbol{x}$ is in occupied space, and 0 otherwise. This information can be obtained from the TSDF by testing the sign of the field at any given point (negative if occupied, positive otherwise). The signed distance invariant, $D_r(\boldsymbol{p})$, is simply the integral of the signed distance field within the bounding sphere of radius r. Formally,

$$V_r(\boldsymbol{p}) = \int_{\boldsymbol{p}+rB} 1_D(\boldsymbol{x})d\boldsymbol{x}, \quad (11)$$

$$D_r(\boldsymbol{p}) = \int_{\boldsymbol{p}+rB} dist(\boldsymbol{x}, \Phi)d\boldsymbol{x} \quad (12)$$

where B is the unit ball. The features are illustrated in

TABLE I

FILTER COEFFICIENTS USED TO DERIVE 3D DERIVATIVE KERNELS

name	derivative	filter
$SoFe_3$	$\begin{bmatrix}1 & 0 & -1\end{bmatrix}^T/2$	$\begin{bmatrix}1 & 2 & 1\end{bmatrix}^T/4$
$SoFe_5$	$\begin{bmatrix}1 & 2 & 0 & -2 & -1\end{bmatrix}^T/6$	$\begin{bmatrix}1 & 4 & 6 & 4 & 1\end{bmatrix}^T/16$
$Scharr_3$	$\begin{bmatrix}1 & 0 & -1\end{bmatrix}^T/2$	$\begin{bmatrix}46.84 & 162.32 & 46.84\end{bmatrix}^T/256$
$Scharr_5$	$\begin{bmatrix}21.38 & 85.24 & 0 & -85.24 & -21.38\end{bmatrix}^T/256$	$\begin{bmatrix}5.96 & 61.81 & 120.46 & 61.81 & 5.96\end{bmatrix}^T/256$

Fig. 3 and Fig. 3, respectively. The mean curvature of the surface is estimated by computing the difference between the result of the integration (or summation, in the discrete case) and the result which would have been produced if the computation had been carried out on a perfectly planar surface. The following expressions approximately relate the mean curvature of the surface to the respective descriptor value.

$$\tilde{H_v}(\boldsymbol{p}) = \frac{8}{3r} - \frac{4V_r}{\pi r^4} \tag{13}$$

$$\tilde{H_d}(\boldsymbol{p}) = \frac{15D_r}{4\pi r^5} \tag{14}$$

From the above equations we note that while the estimated mean curvature for volume integrals is zero if (and only if) the amount of occupied space is equal to half of the sphere, i.e. it is an affine function with a specific reference point. The equation based on the signed distance integral is simply linear. As such, the signed distance integral relates mean surface curvature to the amount of imbalance in the total positive and negative fields on either side. A downside of not using first-order (gradient) information about the field becomes apparent here, as there is no way to distinguish saddle points from flat surfaces, since the mean curvature is zero in both cases.

III. DETECTOR STABILITY EVALUATION

We are interested in evaluating how repeatable the feature descriptors are in the context of robot mapping. Ideally, a robot could return to a previously visited location, or observe a known object and extract geometric descriptors at the exact same places as before, producing a high number of matching descriptors with high confidence. The ideal setting is generally not the case, however. Among the factors that prevent the acquisition of identical maps are differences in measurements from the sensor, variations in pose estimation when integrating the data, and changes in the alignment of the voxel grid. To simplify our analysis, we will only consider the robustness of the feature detectors with respect to changes in the alignment between the initial pose of the voxel grid relative to the sensor. We shall see that this alone has a substantial impact on repeatability, as it includes both sample aliasing in the grid and anisotropy of the feature detectors.

To ensure that the sensor data and estimated trajectory are not a source of variation, we use a pre-recorded data-set with a globally optimized trajectory [24] and reconstruct the environment using the same volumetric integration strategy as Kinect Fusion [1]. At the start of each reconstruction, we transform the initial pose of the camera relative to the voxel volume by increasing amounts of translation and rotation. At the end of each session, the different types of features are extracted and we count the number of features that remained stable in proportion to the total amount. Defining $\boldsymbol{Q}_s$ to be the set of features locations in the unmodified or source configuration and $\boldsymbol{Q}_t$ to be the set of features locations extracted from the target volume, for which the camera pose was initialized with a transformation $\boldsymbol{T}_0$. Let $\boldsymbol{q}_s \in \boldsymbol{Q}_s$ and $\boldsymbol{q}_t \in \boldsymbol{Q}_t$ denote homogeneous vectors in $\mathbb{R}^3$ and $\boldsymbol{T}_0 \in \mathbb{R}^{4\times4}$ a transformation matrix including rotation and translation and $|\cdot|$, the cardinality operator. We then define stability as the average between source to target and target to source matches, where a match is determined to have occurred if two features are within $\tau_{match} = 2$ voxels of each-other.

$$\begin{aligned} score = \frac{1}{2(|\boldsymbol{Q}_s| + |\boldsymbol{Q}_t|)} \\ (|\{\boldsymbol{q}_s, argmin_{\boldsymbol{q}_t} \|\boldsymbol{q}_s - \boldsymbol{T}_0^{-1}\boldsymbol{q}_t\| < \tau_{match}\}| \\ + |\{\boldsymbol{q}_t, argmin_{\boldsymbol{q}_s} \|\boldsymbol{T}_0\boldsymbol{q}_s - \boldsymbol{q}_t\| < \tau_{match}\}|) \end{aligned} \tag{15}$$

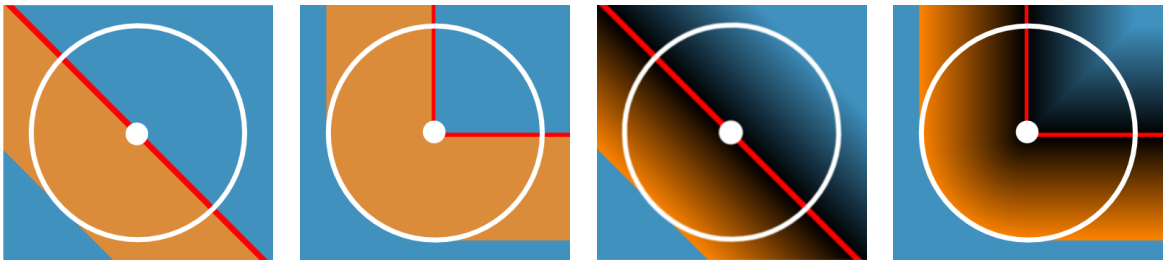
Our definition of the matching score thus avoids being overly generous or strict in case the amount of features differ between the two sets by checking for corresponding features in both directions. We compute the matching scores for varying baselines in translation and rotation:

- translations offsets of $\frac{1}{8}, \frac{2}{8}, \ldots 1$ voxels are applied combinatorially along all dimensions. The sub-voxel shifts are justified by the fact that translating the volume by whole voxel increments does not alter the aliasing and sampling issues that we wish to investigate.
- rotational offsets of $\frac{1}{8}\cdot\frac{\pi}{4}, \frac{2}{8}\cdot\frac{\pi}{4}, \ldots \frac{\pi}{4}$ degrees are also applied combinatorially, around each principal axis. The reason for the chosen interval is that all the algorithms involved are symmetric along the principal axes. Any larger rotations than $\frac{\pi}{4}$ could therefore be achieved by a smaller one and a transposition of the appropriate dimensions (which would not affect the results).

The repeatability score is computed for each reconstruction and descriptor and binned together by the offset relative to the default pose. For translations, we quantify the offset by the L_1 norm. For rotations, we compute the equivalent angle-axis parametrization and bin the results by the magnitude of the angle.

IV. SPARSE STABLE SCENE GRAPHS

As an example application, we present the Sparse Stable Scene Graph. It is a graph structure that uses the features extracted from the TSDF as nodes, and connects a pair of nodes only if the edge doing so is embedded in a surface throughout its length.



(a) (b) (c) (d)

Fig. 3. Integral invariant features: volume invariant in (a),(b), with regions in which the indicator function $1_D(\boldsymbol{x})$ would return 1 (orange) and 0 (blue); distance invariant (c),(d) at a flat surface (zero value) and a corner (negative invariant).

The proposed graph structure can be seen as related to a broader class of representations used for model-based robot vision known as *relational graphs* [25][26]. While relational graphs typically incorporate more semantic meaning in the nodes, we remain on a lower level of abstraction from the data, focusing on geometrically linked points of interest.

In this section we will outline an efficient GPU-amenable way of building the Sparse Stable Scene Graph from a set of feature points, extracted from a TSDF. This method assumes no specific feature point detection method, but requires features to be computed at or very close to the surface. For a given a set of feature points, their fully connected graph can be expressed as a matrix that relates an edge index (the entries in the matrix) to its two endpoint nodes (represented as the row and column index of that entry) e.g. in the following matrix, edge number 7 connects feature points indexed by the numbers 5 (the row) and 2 (the column).

$$\boldsymbol{G} = \begin{bmatrix} - & - & - & - & \cdots \\ 0 & - & - & - & \cdots \\ 1 & 2 & - & - & \cdots \\ 3 & 4 & 5 & - & \cdots \\ 6 & \boxed{7} & 8 & 9 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{16}$$

In practice, one can determine the *zero-based* row and column indices directly from the triangular root of the edge index i_e as,

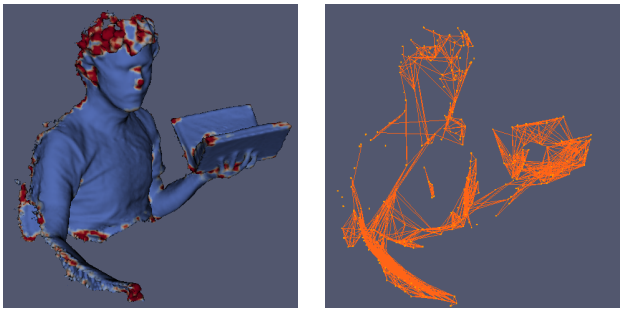
$$row = \lfloor \frac{\sqrt{8i_e + 1} - 1}{2} \rfloor + 1 \tag{17}$$

$$s = \frac{row(row-1)}{2} \tag{18}$$

$$col = i_e - s \tag{19}$$

without actually having to build the matrix. We know in advance that for n_f features there will be exactly $n_e = \frac{n_f(n_f-1)}{2}$ edges in the fully connected graph.

To prune the graph such that it only contains the edges embedded in the surface, we launch n_e separate threads on a GPU. Each thread is designated an index corresponding to the edge index i_e and it is then straightforward to retrieve the feature points referenced by the row and column index using equations (17) and (19). By linearly interpolating between the endpoints of the feature locations we can query the TSDF at a number of points along an edge and reject it if the minimum absolute-valued distance measured along it is above a chosen threshold. The number of points along the edge to test can be made dependent on the length of the edge or constant, if higher accuracy is desired for shorter edges. The pass or fail decision is stored in a binary device vector of the same size as the number of edges and a standard stream compaction [27] operation can then be applied to extract the pruned graph. An illustration of the process is shown in Fig. 4.



(a) (b)

Fig. 4. Computing the SSSG of a scene. Starting from a scene reconstruction, we compute the feature response function (a) and extract feature point locations. After connecting these into a graph and applying our in-surface edge pruning method, enabled by the TSDF, we obtain the final graph (b)

There are a number of possible applications in which the proposed SSSGs can be useful: ranging from place recognition for loop closing in SLAM, through global registration methods, to 3D geomety-based object detection. While in most of these it would be beneficial to also compute a local space feature descriptor in each graph vertex, some sense of the utility of SSSGs can be obtained even without resorting to feature descriptors. We leave further feature-aware SSSG extensions as a future work and instead present a simple descriptorless proof of concept SSSG matching algorithm.

The basic idea of the SSSG matching algorithm is ilustrated in Algorithm 2. In essence, we first filter out the bottom ninety percentile shortest edges, and then generate all possible match combinations of the remaining edges. The match candidates are then filtered to remove edges of widely different lengths, as well as edges whose corresponding vertices have very different degrees. We then run RANSAC on the remaining combinations and check for inliers among all vertices of degree one or higher. The resulting matching algorithm is evaluated in Section V-C.

Algorithm 2 RANSAC-based SSSG matching algorithm

Require: $G_1 = \langle E_1, V_1 \rangle$, $G_2 = \langle E_2, V_2 \rangle$

1: $E_1^f, E_2^f \leftarrow$ top 10 percentile length edges
2: $M \leftarrow$ combinations of E_1^f, E_2^f
3: filter M for unlikely correspondences
4: return $T \leftarrow$ RANSAC best fit transform

V. Experimental Results

The following analysis is based on volumetric integrations of the *copyroom*, *lounge* and *stone wall* data-sets [1] with SDF volume size of 512^3 voxels with varying voxel sizes and a truncation distance of $\pm v_{size} * 4$. The relatively large truncation distance is chosen to give the integral invariant features a better chance at producing stable features; see the discussion in sec. V-B. For each reconstruction session we use the first 2500 frames, as this captures representative parts of the scene with diverse characteristics. For all experiments, we set the non-maxima suppression window to be of size $7 \times 7 \times 7$ and set thresholds for culling features for which the response function is low. The saliency detectors and SSSG extraction procedures were implemented using CUDA and deployed on an NVIDIA GeForce GTX Titan GK110 GPU with 6Gb of memory. A single-core CPU-space program was used for interfacing and deployed on a computer with and Intel Core i7 CPU at 3.50GHz.

A. Gradient-based methods

We find that Harris features are generally more stable than Shi & Tomasi features and as we decrease the proportion of features that we keep, by increasing the rejection threshold, their performances become similar. Over all of the experiments reported here, the stability scores of the Shi & Tomasi detectors were on average equal to 91% of those of the corresponding Harris detectors, and thus for presentation reasons we will omit their curves from the following plots. We show the performance of Harris features, when computed using different gradient estimation methods in Fig. 5. The smaller $3 \times 3 \times 3$ kernels all produced slightly worse results than their larger counterparts though with similar trends, we omit them for clarity of presentation. We find that the derivative of Gaussian outperforms Sobel-Feldman which in turn outperforms Scharr kernels. For all three, increasing the rejection threshold results in a larger proportion of stable features.

Central differences are cheaper to compute but offer poor repeatability, and when the feature rejection threshold is increased a larger proportion of high quality features are culled, noted by the drop in repeatability.

The sensitivity with respect to rotation is shown in Fig. 5(c). Note that since the volume is not pivoted around the feature locations but around the camera origin, some translation is induced as well, explaining why the curves do not begin at 1. Although the Scharr kernel produces the least amount of variation with respect to rotation, the repeatability of Harris features is higher when gradients are computed based on both Sobel-Feldman and derivative of Gaussian kernels. Central differences provide the least robust gradient estimate, under rotation, as expected.

In all cases , the window size w over which the summation of the gradients to form $\boldsymbol{H}$ (see Eq. (3) is carried out was set to $5 \times 5 \times 5$.

B. Integral-based methods

The integral based methods, namely the volume integral and signed distance integral features, do not perform as well as the gradient-based method. We see in Fig. 5(a)—5(b) that their repeatability is below that of all variants of Harris features. Increasing the rejection threshold does not produce much improvement for the signed distance integral, and causes a slight deterioration in the case of volume integrals. The rotation invariance, shown in Fig. 5(c) is good, in spite of the spherical integration region being a discrete approximation with a radius of 3.5 voxels. The diameter of the integration region is made to match the truncation distance of ± 4 voxels. However, the actual width of the non-truncated region of the TSDF may both be larger, depending on sensor noise, and smaller, due to surfaces being at grazing angles relative to the line of sight to the sensor. The dependency between integration radius and truncation distance is more critical for volume invariants, where the signed distance field should ideally not be truncated within the radius of the integration region. This is because the volume integral is compared to a specific reference value for curvature estimation, and this reference would need to be readjusted in case half of the volume being occupied, as seen in Fig 3(a) no longer corresponds to a planar surface.

Additionally, for signed distance invariants the thickness of the thinnest object should be at least twice as large as the radius of the integration region. This is due to the balance needed between positive and negative fields to indicate a mean curvature of zero. In the case of thin objects, the negative side of the distance field will only decrease until the mid-section of an object leading to skewed estimates.

To highlight another problem of integral invariants applied to truncated distance fields, consider Fig. 6. Indicated in green is the amount of additional occupancy caused by a sharp concave bending of the surface. Here we see that increasing the radius of the integration region beyond the truncation distance adds no useful information to the curvature estimation. In fact, it only serves to reduce the relative difference between curved and planar surfaces, compare the case of Fig. 6 with those of Fig. 3 for example. In the latter, the relative difference between the planar, Fig. 3(a), and curved, Fig. 3(b), case is large; changing from $\frac{1}{2}$ to $\frac{3}{4}$ of the circle area.

The results show that the signed distance integrals fare slightly better than the volume integrals which is expected since the former uses both the positive and negative regions of the field and therefore has a slightly larger sample base. However, neither is large enough to robustly filter out noise at the tested radius. Since merely extending the radius is

[1] available from http://qianyi.info/scenedata.html

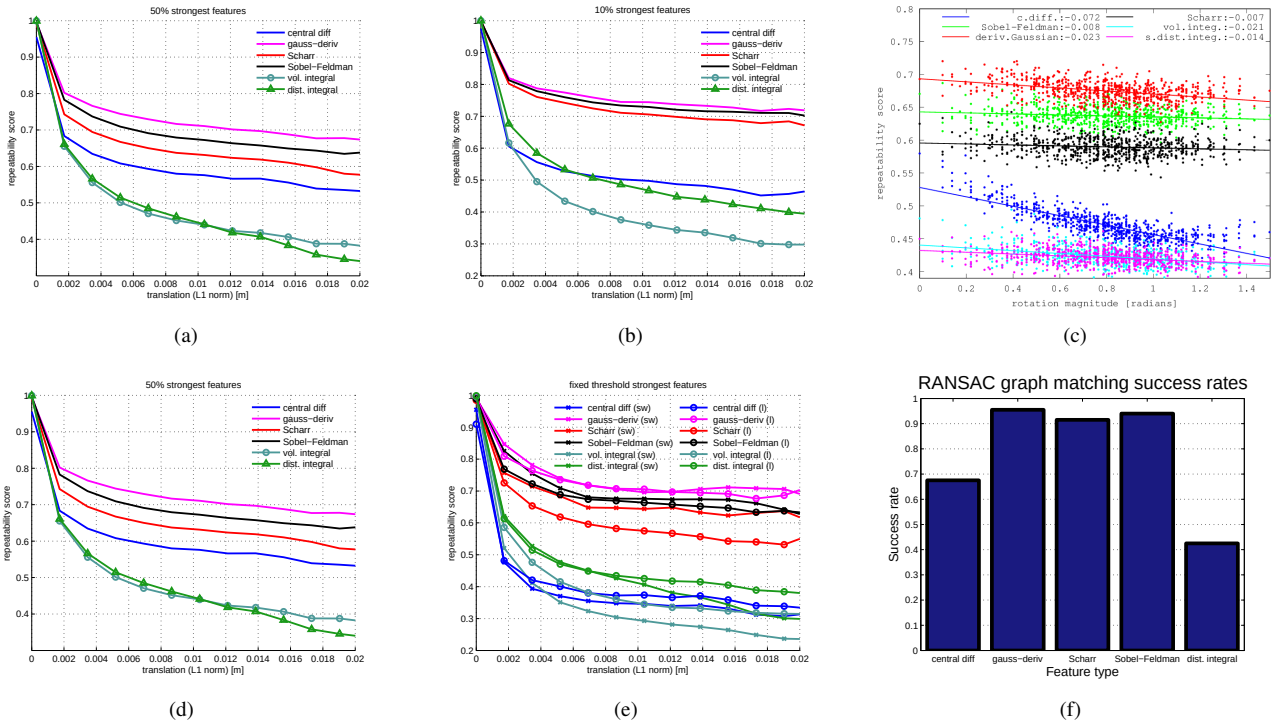


Fig. 5. Repeatability of Harris features computed with the derivative of Gaussian kernel (size $5 \times 5 \times 5$) when threshold is set such that only the top 50% and 10% features are maintained

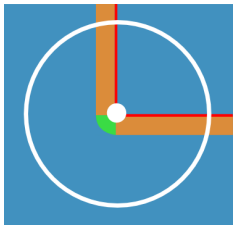


Fig. 6. 2D analogy of volume integral with occupancy estimated on a narrowly truncated signed distance field. The green region shows the total additional 'volume' that appears due to a sharp right-angled bend

fruitless, we are left with the option of extending the TSDF truncation distance to accommodate a larger integration region. However, this leads to a decrease in the quality of scene reconstruction and makes the reconstructed geometry more dependent on the sensor's viewpoint locations.

The main advantage of integral invariant features is that they are relatively cheap to compute, requiring only a sum reduction of the distance field within a bounding region. To make the computational cost more tangible, a naive implementations of derivatives by convolution with a filter kernel take between 680ms and 150ms for kernels of $5\times5\times5$ and $3 \times 3 \times 3$ respectively. Computing the Harris features adds another 100ms, with approximately 8ms more for the non-maxima suppression. In contrast, the integral invariants take around 20ms to compute, followed by the non-maxima suppression adding to a total of 28ms.

C. Matching Sparse Stable Scene Graphs

The graph generation is roughly equivalent to a simple ray casting operation, though typically with fewer and shorter rays than needed to render a VGA video frame from the TSDF. An edge can also be rejected a priori, if its endpoints are deemed to be too far apart. This rejection criterion promotes the formation of more numerous disjoint graphs which tend to have a closer relationship to objects than overall scene structure. Any number of segmentations based on edge lengths may be chosen if meaningful heuristics are available for the particular environment. Typical timings obtained are 2ms, 8ms, 26ms for 260, 700 and 1200 features, respectively.

Our method for building Scene Graphs is very efficient, though the resulting graph is only as stable as the features used to generate it. Occasionally, stable features fail to be connected because they appeared at some distance away from the surface. This can be potentially be mitigated by shifting the features in the direction of the local surface, i.e., for each feature point $\boldsymbol{q}$, compute an updated $\boldsymbol{q}' = \boldsymbol{q} - \alpha\ dist(\boldsymbol{q}, \Phi)(\Delta dist(\boldsymbol{q}, \Phi))$, with α parametrizing the size of the shift. Since corner points are, by definition, the regions where gradient variation is high, the final location of $\boldsymbol{q}'$ may be difficult to predict.

VI. Summary and Conclusions

Our experiments on integral invariant features suggest that the zero-order information contained in local regions of a discretely sampled TSDF is not enough to separate truly salient regions from noise and that gradient-based methods perform better. It is possible that machine learning methods such as FAST-ER[28] that builds a decision tree optimized for finding repeatable points may perform better even in the absence of gradient information.

Requiring gradients does not necessarily bar features from being used in real-time applications since linearly separable

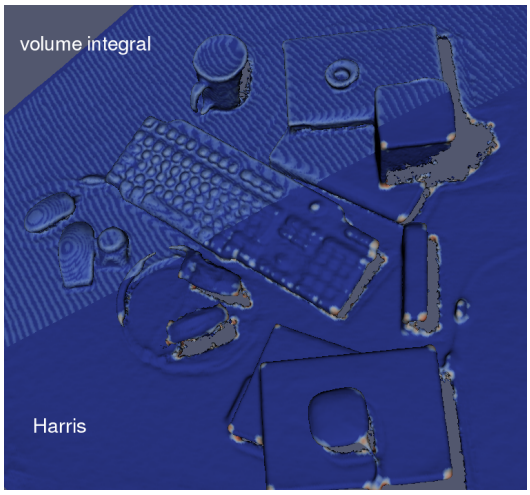


Fig. 7. A tabletop scene with volume integral and Harris features side-by-side

filters can be applied very efficiently, even for large volumetric images. Furthermore, since gradients are typically computed at every frame during the camera-tracking process, these could be incrementally fused into separate volumes over time, as well. Other performance strategies such as empty-space skipping may also be used to make the computation more tractable.

During our experiments we observed some practical characteristics of the feature detectors, worth highlighting. In Fig. 7 we see the implications of the integral response function not being computed exactly on the surface, and why shifting the grid will have an impact on the results. As explained before, the integral invariants erroneously increase their response as the integral is evaluated further away from the surface and should ideally only be computed exactly on it. We have also observed that Harris corner maximum responses tend to occur closer to the center of a corners radius rather than the surface, since this is where gradients are both different and unmixed.

We have shown that feature detection in 3D, coupled with the TSDF representation can support novel applications, such as the SSSG. Whether this graph structure itself is a useful approximation to the real map in a SLAM scenario remains to be investigated. However, it is unquestionably a lightweight wire-frame representation that can be used for remote visualization, comprised of several orders of magnitude fewer elements than the polygons produced by e.g. marching-cubes [29].

REFERENCES

[1] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, 2011, pp. 127–136.

[2] H. Roth and M. Vona, "Moving volume kinectfusion." in *BMVC*, 2012, pp. 1–11.

[3] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul 2012.

[4] D. R. Canelhas, T. Stoyanov, and A. J. Lilienthal, "Sdf tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3671–3676.

[5] O. Kahler, V. Prisacariu, C. Ren, X. Sun, P. Torr, and D. Murray, "Very high frame rate volumetric integration of depth images on mobile devices," *Visualization and Computer Graphics, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[6] B. L. Curless, "New methods for surface reconstruction from range images," Ph.D. dissertation, Stanford University, 1997.

[7] J. C. Hart, "Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces," *The Visual Computer*, vol. 12, no. 10, pp. 527–545, 1996.

[8] D. R. Canelhas, T. Stoyanov, and A. J. Lilienthal, "Improved local shape feature stability through dense model tracking," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3203–3209.

[9] T.-H. Yu, O. J. Woodford, and R. Cipolla, "A performance evaluation of volumetric 3d interest point detectors," *International journal of computer vision*, vol. 102, no. 1-3, pp. 180–197, 2013.

[10] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Citeseer, 1988, p. 50.

[11] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE, 1994, pp. 593–600.

[12] H. Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang, "Integral invariants for robust geometry processing," *Computer Aided Geometric Design*, vol. 26, no. 1, pp. 37–60, 2009.

[13] C. H. Lee, A. Varshney, and D. W. Jacobs, "Mesh saliency," in *ACM transactions on graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 659–666.

[14] I. Pratikakis, M. Spagnuolo, T. Theoharis, and R. Veltkamp, "A robust 3d interest points detector based on harris operator," in *Eurographics Workshop on 3D Object Retrieval*, vol. 1. Citeseer, 2010.

[15] A. Godil and A. I. Wagan, "Salient local 3d features for 3d shape retrieval," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2011, pp. 78 640S–78 640S.

[16] H. Dutagaci, C. P. Cheung, and A. Godil, "Evaluation of 3d interest point detection techniques via human-generated ground truth," *The Visual Computer*, vol. 28, no. 9, pp. 901–917, 2012.

[17] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 303–312.

[18] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover." DTIC Document, Tech. Rep., 1980.

[19] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 650–663.

[20] I. Sobel and G. Feldman, "A 3x3 isotropic gradient operator for image processing," 1968.

[21] E. Dam and B. ter Haar Romeny, "Front end vision and multi-scale image analysis," *Deep Structure I, II & III*, no. 1-4020, pp. 1507–0, 2003.

[22] H. Scharr, "Optimal operators in digital image processing," Ph.D. dissertation, 2000.

[23] S. Manay, B.-W. Hong, A. J. Yezzi, and S. Soatto, *Integral invariant signatures*. Springer, 2004.

[24] Q.-Y. Zhou and V. Koltun, "Dense scene reconstruction with points of interest," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 112, 2013.

[25] R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision," *ACM Computing Surveys (CSUR)*, vol. 18, no. 1, pp. 67–108, 1986.

[26] R. Nevatia and T. O. Binford, "Description and recognition of curved objects," *Artificial Intelligence*, vol. 8, no. 1, pp. 77–98, 1977.

[27] J. Hoberock and N. Bell, "Thrust: A parallel template library," *Online at https://thrust.github.io*, vol. 42, p. 43, 2010.

[28] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 105–119, 2010.

[29] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM siggraph computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 163–169.