TRISK: A local features extraction framework for texture-plus-depth content matching

Maxim Karpushin, Giuseppe Valenzise, Frederic Dufaux

TRISK: A local features extraction framework for texture-plus-depth content matching

Maxim Karpushin[a], Giuseppe Valenzise[b,*], Frédéric Dufaux[b]

[a]*LTCI, Télécom ParisTech, Université Paris-Saclay, 46 rue Barrault, 75013, Paris, France*
[b]*L2S, CNRS, CentraleSupelec, Université Paris-Sud, 3 rue Joliot Curie, 91192, Gif-sur-Yvette, France*

Abstract

In this paper we present a new complete detector-descriptor framework for local features extraction from grayscale texture-plus-depth images. It is designed by putting together a locally normalized binary descriptor and the popular AGAST corner detector modified to incorporate the depth map into the keypoint detection process. With these new local features, we target image matching applications when significant out-of-plane rotations and viewpoint position changes are present in the input data. Our approach is designed to perform on RGBD images acquired with low-cost sensors such as Kinect without any complex depth map preprocessing such as denoising or inpainting. We show improved results with respect to several other highly competitive local image features through both a classic local feature evaluation procedure and an illustrative application scenario. Moreover, the proposed method requires low computational effort.

Keywords: texture-plus-depth, RGBD, local feature, keypoint detector, descriptor, viewpoint changes

1. Introduction

During the past decades, a large spectrum of vision problems has been settled with local features, such as visual simultaneous localization and mapping (SLAM) [1], visual odometry [2], tracking by matching [3], etc. This has

[*] Corresponding author
Email address: giuseppe.valenzise@l2s.centralesupelec.fr (Giuseppe Valenzise)

made the concept of local features one of the most valuable in vision. Numerous comparative evaluations of competing local features have been published [1, 4, 5, 6, 7, 8, 9, 10, 11]. Industrial demand for universally applicable local image features has also stimulated MPEG standardization activities for Compact Descriptors for Visual Search (CDVS) [12] and Compact Descriptors for Visual Analysis (CDVA) [13].

Intensive development of local features in traditional imaging has nowadays arrived to the exploration of different visual content modalities, such as range images, 3D meshes or plenoptic images. This is further stimulated by the commercial diffusion of the corresponding acquisition devices, such as low-cost RGB+depth sensors Microsoft Kinect, ASUS Xtion, Google Tango, Structure Sensor for iPad, high quality laser scanners (LIDARs), lightfield cameras Lytro, Raytrix, etc. In this work we consider the *RGBD* format, also known as *"texture-plus-depth"*, in which a conventional 2D image (*texture map*) is complemented by a range image (*depth map*) describing the distance of objects from the camera plane[1].

Recently, a good deal of attention has been devoted to designing novel local features for RGBD content. In fact, differently from 3D meshes and point clouds, this modality allows to employ and extend principles of local features from traditional imaging. However, in spite of this growing interest, to the best of our knowledge no complete feature extraction pipeline (containing both *detector* and *descriptor*) has been proposed so far for (sparse) RGBD local features. This has been partially due to the noisiness and incompleteness of depth maps acquired by low-cost sensors such as the Kinect.

In this paper, we show that the geometrical information provided by depth, if properly used, enables to improve the stability of local features harnessed from texture images. Especially, feature invariance to rigid 3D transformations, which is the most common class of visual deformations, may be significantly

[1] In this paper we do not deal with the color aspect, so in what follows by RGBD we mean "grayscale-plus-depth".

increased. This is of high practical interest as out-of-plane rotations are known failure cases for classic texture-only local features. In addition, we demonstrate that the proposed features can be computed efficiently.

The contribution of this paper is a new local features extraction framework for RGBD (texture-plus-depth) sparse image matching that consists of: *i)* a salient visual point detector based on a corner detector, and *ii)* a binary local feature descriptor. Differently to other state-of-the-art RGBD local features, in our approach the depth map is involved in both stages. Moreover,

- the proposed feature is designed to be robust to viewpoint position changes, whereas all the standard state-of-the-art feature invariance classes (translations, in-plane rotations, scale changes, simple illumination changes) are preserved;

- our method is applicable to real RGBD data of Kinect quality taking into account the major flaws of the D channel. We only assume that the depth map is aligned with the texture map through a device-specific camera calibration transformation, which is typically provided with the sensor;

- feature detection and description require a moderate computational effort and are easily parallelizable. The resulting descriptors are binary, allowing for extremely fast matching.

The rest of the paper is organized as follows. Section II presents related work on local features and introduces the problem of out-of-plane rotations. Section III describes the design of the proposed feature extraction pipeline. Section IV presents in details the experimental validation and obtained results. Finally, Section V concludes the paper.

2. Related Work

2.1. Conventional local features

The idea of content matching through local features has been progressively evolving for a long time, but the concept of a robust universal local image feature, i.e., a feature designed regardless of a specific application, is relatively modern. Sparse image matching through such features typically consists of three steps:

- **detection** of repeatable salient visual points (*keypoints*) in the input image,
- **description**: computation of a compact signature (*descriptor*) describing locally the visual content at each keypoint detected on the previous stage,
- **matching**: for two given images each represented by a set of such descriptors, establishing pairwise correspondences between the feature sets revealing local visual similarities.

The number of the correspondences, their fidelity and the underlying geometry are then analyzed by the application in order to decide on the similarity of the input images in search tasks, or to figure out the geometrical relation between two views in localization and registration tasks.

SIFT (Scale Invariant Feature Transform) [14] was the first complete and universal framework to detect keypoints and extract corresponding local descriptors that are scale and rotational invariant. *SURF* (Speeded Up Robust Features) [15] was then proposed as a computationally efficient alternative to SIFT. Both approaches use pyramidal image representations to detect scale invariant keypoints, and describe the surrounding patches by high-dimensional histogram-based signatures. The matching of such descriptors relies on the Euclidean distance.

More recently, a greater deal of attention has been devoted to *binary local features*: they increase the computational efficiency of feature extraction and matching, and together with learning-based approaches are currently an active research field in the computer vision community [16, 17, 18, 19, 20, 21, 22]. One of the first proposed binary features, *BRIEF* (Binary Robust Independent Elementary Feature) [23] extends the idea of local binary patterns [24], originally designed for texture analysis tasks, to describe interesting points. Since the extracted feature is a string of bits, the matching is done using Hamming distance, which is more efficient to compute than the Euclidean one. This idea is further elaborated in numerous works [25, 26, 27, 28, 29]. Notably, *ORB* (Oriented FAST and Rotated BRIEF) [25] and *BRISK* (Binary Robust Invariant Scalable Keypoints) [26] present complete extractors of scale and rotation invariant binary features. They apply *FAST* [30] and *AGAST* [31] corner detectors to scale space-like image pyramids to find the keypoints, estimate dominant keypoint orientations, and then invoke the same principle of binary description. The feature proposed in this work employs a similar binary pattern, but we sample it in the scene surface rather than in the camera plane.

2.2. The problem of out-of-plane rotations

Existing 2D scale and rotational invariant features are not suited to deal with considerable 3D distortions, even rigid, such as perspective deformations, rotations out of the camera plane, or substantial camera position changes. As an example, SIFT performance drops quickly when the scene undergoes an out-of-plane rotation of more than 45° [32]. According to different evaluations [4, 6, 9], this trend is common to most detectors and descriptors. For this reason, a set of approaches dealing with such 3D distortions has been developed.

Affine invariant features address the problem assuming that perspective distortions are well approximated locally by in-plane affine transformations. Affine-covariant detectors [33] estimate an elliptical frame per keypoint using the surrounding image content. The local patch then undergoes a normalizing transformation mapping each estimated ellipse to a circle. *ASIFT* (Affine-SIFT) [32] is based on an alternative paradigm, i.e., it *simulates* a set of affinely transformed versions of the input image in order to find the best matching features. A similar simulation-based affine generalization of SURF is presented in [34]. Some approaches go beyond the rigid scene deformations, aiming at non-rigid surfaces images matching, e.g., movement of textiles [35, 36].

An essential limitation of the affine invariance paradigm is that perspective distortions are approximated by a class of transformations that is too general.

This causes losses of relevant visual information. A typical example is that affine-covariant features do not distinguish between a square and a rectangle, or a circle and an ellipse [37]. As we showed in our previous work [38], this leads to a loss of the descriptor discriminability.

While the invariance of conventional features, such as SIFT or BRISK, to translations, scale changes and in-plane rotations is guaranteed by design, the invariance to out-of-plane rotations of the listed approaches is rather heuristic. This leads to limited feature stability when the observed scene undergoes significant viewpoint position changes. Therefore, *out-of-plane rotations* and *viewpoint position changes* still remain challenging. We consider these two transformation classes as synonyms in the following, since combined with translations, scale changes and in-plane rotations they become equivalent to 3D rigid scene deformations. The problem of feature invariance is thus the focus of this paper: we believe that the main advantage of injecting complementary geometrical information into the feature extraction process is the possibility to deal with significant viewpoint position changes.

2.3. Texture+Depth (RGBD) content matching

A considerable amount of work has been done on the local features for range images (depth maps) as well as RGBD images. Such methods may be split into three groups.

Shape-only descriptors. Some local descriptors operate only with depth maps or point clouds. These approaches are advantageous in applications where the geometrical information is prevalent over the photometrical one. Absence of texture in the feature computation process makes the features completely insensible to any kind of illumination changes. However, in case of poorly detailed geometry the performance of such approaches drops off. *2.5D SIFT* [39] proposes an extension of SIFT detector and descriptor to range images. *NARF* (Normally Aligned Radial Feature) [40] is a rotational invariant feature detector and descriptor for range image matching. *SIPF* (Scale Invariant Point

Feature) [41] is a recent work on the detection and description of scale invariant keypoints in point clouds. Other descriptors for shape matching are proposed in [42, 43, 44, 45, 46, 47].

Joint shape-texture description. In the second case, shape and texture are described jointly, i.e., a signature at each interest point describes both the local geometrical and photometrical information simultaneously. Joining the two modalities allows for improved robustness of detected features in static environments, e.g., for indoor localization. *CSHOT* (Color SHOT) [48] and *BRAND* (Binary Robust Appearance and Normal Descriptor) [49] propose binary descriptors obtained by properly combining two separate signatures extracted at the same keypoint from the texture map and the depth map. None of these methods, however, deals with significant viewpoint position changes.

Texture description using shape. In the third and last case, the geometry may be used to provide a robust description of the texture, but is not explicitly incorporated into the resulting descriptors. Differently to the previous case, such techniques are based on texture characteristics that are invariant with respect to the local shape. In this way a consistent deformation of the observed scene that affects both texture and geometry does not impact the descriptor. This reveals a particular interest for invariance to out-of-plane rotations. *VIP* (Viewpoint Invariant Patches) [37], *PIN* (Perspectively Invariant Normal features) [50], *DAFT* (Depth-Adaptive Feature Transform) [51] and our previous work [38] present descriptor patch normalization techniques aimed at improved stability under significant viewpoint position changes. The latter three perform a local normalization approximating the scene geometry near each keypoint by a plane, and then properly transforming the descriptor patch. VIP proceeds in a more global way. It looks for several dominant planes in the scene, then synthesizes corresponding frontal views and computes their SIFT descriptors. In our preliminary work [38], we computed a simple least-square local planar warping of the texture surface in order to *deslant* it before computing a blob or corner descriptor. Differently to that work, here we directly sample the keypoint and the descriptor patterns in the local axes in the camera plane, which turns out to be computationally more efficient. Our recent work [52] presents a technique allowing more repeatable and distinctive BRISK features from the texture image by mapping the intensity sampling pattern onto the scene surface. However, that approach has the main limitations of computational complexity and sensitivity to noise, which has motivated us to turn towards a locally planar and faster pattern-to-surface mapping algorithm in this paper. Some approaches for mesh matching may be considered in the same context, such as *MeshDOG+MeshHOG* (Difference of Gaussians + Histogram of Oriented Gradients) [53]. However, they require additional preprocessing steps to render a proper mesh from an RGBD image, whereas MeshDOG itself is already quite computationally expensive.

RGBD scale-invariant keypoint detection. In [54, 55, 56] we focus on the problem of keypoint detection for RGBD. Specifically, we proposed a scale space formulation for the texture image that exploits the surface metric given by the depth map, by means of a Laplacian-like operator defining a non-uniform diffusion process [54]. In a follow-up work [56] we have employed this operator to conceive a complete multi-scale RGBD blob detector. While that work is mathematically elegant, it has the disadvantage of being computationally complex, as it entails performing an anisotropic diffusion process. In this work, we consider instead highly performing binary features, for which we do not need to compute derivatives explicitly.

3. TRISK: The Proposed Method

3.1. Overview

In this section we present the design of a keypoint detector and a feature descriptor for RGBD image matching. Our final goal is to obtain reliable features under significant viewpoint position changes, which are robust to depth map imperfections and at the same time computationally efficient. As briefly discussed in Section 2, visual features have been vastly studied for many years, leading to a number of tools that have been proven successful for image matching. We build on this knowledge base and retain the best concepts formulated so far, but we rethink and adapt them to introduce the geometric information provided by RGBD content.

In particular, we consider as a starting point the popular BRISK features [26], which provides state-of-the-art performance in both feature quality and computational speed amongst binary features [8]. However, our framework is rather general in principle and could be equally applied to other binary features. To underline the continuity with the visual feature literature and specifically BRISK, we then call the proposed features **TRISK**, for "Tridimensional Rotational Invariant Surface Keypoints". The overall scheme of TRISK is shown in Fig. 1. In the rest of this Section we describe in detail the building blocks of the proposed detector/descriptor.

3.2. The Detector

The proposed feature extraction algorithm begins with the following steps.

3.2.1. Local surface axes computation

The goal of this work is to render the feature extraction process as independent as possible of the camera position. One way to do this is to adapt all the local processing to the surface geometry, considering the observed image as a textured manifold. In TRISK, we follow this way by selecting a proper basis at each image point, which we further refer to as *adaptive local axes.* They are used to transfer the detection and the description from the camera plane onto the scene surface, basing them on the surface metric, which is intrinsically independent of the reciprocal camera-to-object position and orientation.

Deriving the adaptive local axes from the depth map is at the base of TRISK. The following local operations will then be performed in the derived local axes: AGAST corner score computation, Harris cornerness test, accurate keypoint localization and descriptor sampling. We first explain the proposed technique

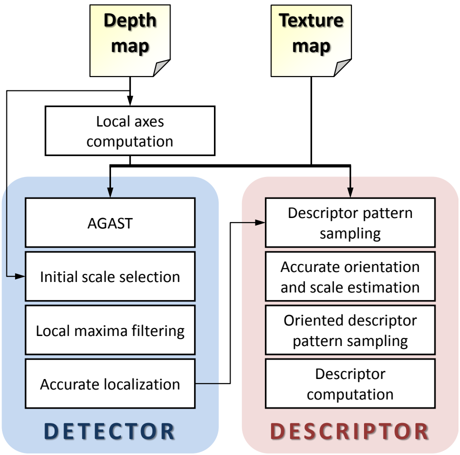

Figure 1: The architecture of the proposed TRISK pipeline. TRISK is a complete feature extraction framework for RGBD content, composed by a keypoint detector and a descriptor. Both leverage the geometric information provided by the depth map in order to sample the texture considering a different local coordinate system for each point of an object surface. The *detector* is based on the Adaptive Generic Accelerated Segment Test (AGAST) response, computed in local coordinates. Depth is also used to find the approximate *geometric* scale of a keypoint, which is further refined at the description stage together with orientation normalization. The local maxima filtering and accurate localization stages enable to select the most repeatable keypoints. In order to compute the *descriptor*, the texture is sampled again in local coordinates. A multi-pass procedure is employed to accurately estimate the orientation and scale of the sampling pattern. Finally, similarly to the BRISK descriptor, pairwise comparison tests across the texture samples are carried out to produce a binary descriptor string.

to compute the local axes and then present the details of their use in the feature extraction.

Assuming that keypoint detection and description are rotationally invariant, the local axes are given by any orthonormal basis of the tangent plane, projected on the camera plane and normalized so that its largest vector has unit norm in pixels. Examples are shown in Fig. 3. In the following we derive an analytic expression of the local axes field, allowing to compute them efficiently at each pixel location.

Let us consider a camera with the centered principal point. According to

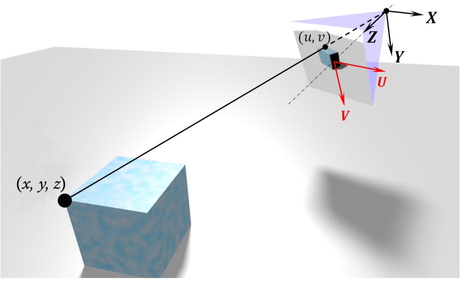

Figure 2: Local camera coordinates (x, y, z) and image plane coordinates (u, v).

the perspective projection model, the relation between a spatial point (x, y, z) and its projection $(u, v) = \mathbf{Proj}\,(x, y, z)$ on the camera plane is then expressed by the following formula (the corresponding coordinate systems are presented in Fig. 2):

$$u = \frac{x}{z},\ v = \frac{y}{z}$$

Let A denote a scene point, $\vec{A}$ its coordinate vector in camera coordinates and $(u, v) = \mathbf{Proj}\left(\vec{A}\right)$. Let $\vec{n} = (n_x, n_y, -n_z)$ be the surface normal of unit norm at A (see Fig 3). With no loss of generality we assume $0 < n_z < 1$.

The following reasoning is based on the observation that the degree of perspective distortions along a contour on the scene surface passing through A depends on its direction with respect to the camera plane. Specifically, a tangent line L parallel to the camera plane is not affected by the perspective distortions: there is no contraction along L when projecting it on the camera plane. Nothing prevents to use this line as the first local axis. Thus, we need to find a vector $\vec{m}_1 = (m_x, m_y, m_z)$ such that it is: i) parallel to the camera plane; and ii) belonging to the tangent plane at A. The first condition results in $m_z = 0$. The second condition requires that $\vec{n} \cdot \vec{m} = 0$. It is straightforward to verify that $\vec{m}_1 = (-n_y, n_x, 0)$ satisfies both conditions. Let $\vec{q'}_1 = \mathbf{Proj}\left(\vec{A} + \vec{m}_1\right) - \mathbf{Proj}\left(\vec{A}\right)$ be the projection of $\vec{m}_1$ onto the camera plane. As there is no contraction along L, we normalize $\vec{q'}_1$ to have *always* unit

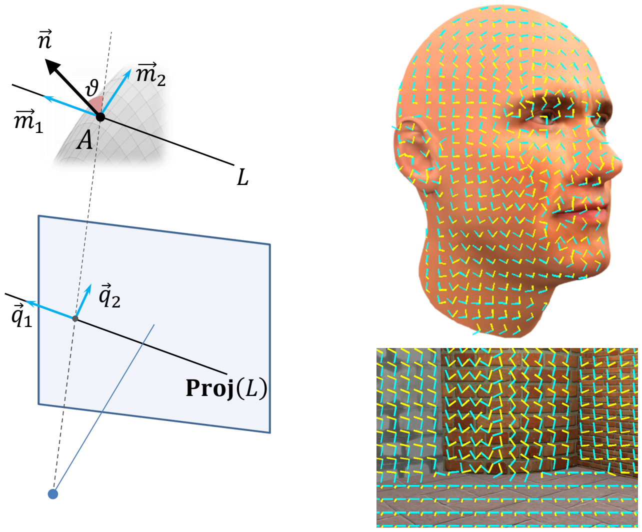

Figure 3: Computation of local axes $\vec{q}_1$ and $\vec{q}_2$. On the left: $\vec{q}_1$ and $\vec{q}_2$ are obtained by projecting $\vec{m}_1$ and $\vec{m}_2$ in the 3D space onto the camera plane. $\vec{m}_1$ is chosen to be always parallel to the camera plane, and its projected local axis is normalized to unit length. The projection of $\vec{m}_2$, i.e., $\vec{q}_2$ has a length reflecting the perspective distortion at A, which depends on the angle ϑ between the viewpoint $\vec{A}$ and the normal at A. On the right: examples of local axes fields computed on images from *Arnold* and *Bricks* sequences, with $\vec{q}_1$ shown in cyan and $\vec{q}_2$ in yellow.

norm, i.e., $\vec{q}_1 = \|\vec{q'}_1\|^{-1}\vec{q'}_1$, obtaining the first local axis as:

$$\vec{q}_1 = \frac{1}{\sqrt{n_x^2 + n_y^2}} \begin{pmatrix} -n_y \\ n_x \end{pmatrix}. \tag{1}$$

The second required spatial vector $\vec{m}_2$ must be orthogonal to both $\vec{n}$ and $\vec{m}_1$, as together with $\vec{m}_1$ it forms an orthogonal basis on the surface. This can be found by the cross product: $\vec{m}_2 = \vec{m}_1 \times \vec{n}$. Along $\vec{m}_2$ distances are contracted by a factor which depends on the cosine of the angle ϑ between the viewpoint vector $\vec{A}$ and the normal $\vec{n}$ (see Figure 3): when $\vec{A}$ and $\vec{n}$ are aligned, then the tangent plane is parallel to the camera plane and there is no contraction; conversely, when $\vec{A}$ and $\vec{n}$ are orthogonal, the distortion is maximal. Let $\vec{q'}_2 = \mathbf{Proj}\left(\vec{A} + \vec{m}_2\right) - \mathbf{Proj}\left(\vec{A}\right)$ be the projection of $\vec{m}_2$ onto the camera plane.

The second local axis is thus given by:

$$\vec{q}_2 = \frac{\vec{A} \cdot \vec{n}}{\|\vec{A}\| \|\vec{n}\|} \cdot \frac{\vec{q'}_2}{\|\vec{q'}_2\|} = \frac{n_x u + n_y v - n_z}{\|\vec{q'}_2\| \sqrt{u^2 + v^2 + 1}} \vec{q'}_2, \tag{2}$$

where

$$\vec{q'}_2 = \begin{pmatrix} \dfrac{n_x n_z - u}{n_x^2 + n_y^2 - 1} - u \\ \dfrac{n_y n_z - v}{n_x^2 + n_y^2 - 1} - v \end{pmatrix}. \tag{3}$$

The derived expressions of $\vec{q}_1$ and $\vec{q}_2$ depend only on the surface normal and the point position on the camera plane (u, v), but not on the depth map values directly. To estimate the normal vector we use PCA-based normal estimation [57]. Since the depth noise increases with the distance for many sensors, including Kinect, we scale the support size with the depth. The scaling factor κ is an input parameter, whose tuning is discussed below. Using this approach the local axes field may be computed in $O(N)$ operations for an input image of N pixels. Moreover, it avoids explicit manipulations with differential characteristics of the depth map, which are prone to noise.

The described technique allows to compute the adaptive local axes from the depth map in a computationally efficient way and robustly to the noise. Under the assumption of the rotationally invariant keypoint detection criteria, this choice of basis vectors is not unique: a simple alternative is to choose the other two vectors obtained by the PCA decomposition. This, however, takes more computational time than the proposed technique (we discuss this option briefly in the experimental part).

3.2.2. AGAST and scale selection

Adaptive Generic Accelerated Segment Test [31] is an approach for corner detection in images. According to this test, a pixel is deemed to be a corner if it is darker or brighter than at least N connected points on a circle surrounding it. More specifically, a pixel in the circle is considered darker/brighter than the center pixel if its intensity value is smaller/larger than the center intensity by at least a value τ. Therefore, keypoint detection with AGAST depends on the choice of τ and N. By increasing the value of τ, a smaller number of corners with progressively increasing contrast are selected. As suggested in [26], in order to obtain a per pixel score and perform non-local maxima suppression as, e.g., in SIFT [14], we define the score $s(i)$ of pixel i as the maximum value of the intensity difference threshold τ such that i passes the AGAST corner test. Intuitively, pixels with higher scores correspond to higher contrast corners, which are likely to be more repeatable. Pixels whose score reaches a local maximum greater than a threshold t are taken as keypoint candidates.

This detection principle was successfully involved in scale-covariant keypoint detection [25, 26]. Due to its isotropic (rotational invariant) and derivative-free design, this detection principle demonstrates good stability to image noise and moderate geometric deformation. In our case, the isotropic detection is required for using local adaptive axes. Moreover, AGAST allows to save time by reducing the number of intensity comparisons using a properly learned decision tree. This also responds well to our needs, since the image interpolation in the local axes is time consuming.

Specifically, inspired by BRISK detector [26], we apply AGAST to pick the keypoint candidates as explained below.

AGAST in local axes. Aiming at improved stability to viewpoint position changes, we apply AGAST9-16 in the local adaptive axes (“9-16” stands for at least 9 darker or brighter pixels on a circle of 16 pixels). The texture map is interpolated using the local surface axes defined in Eq. 2. Let us consider a Bresenham circle, i.e., a discrete approximation of a circle with N points (in our case, $N = 16$). Let $\{(u_k, v_k)\}_{k=1}^{16}$ be the coordinates of the points of that circle, where the reference system has origin in the center pixel. In order to transform the set of vectors $\{(u_k, v_k)\}_{k=1}^{16}$ from this coordinate system into vectors $\{(x_k, y_k)\}_{k=1}^{16}$ expressed in the local axes $(\vec{q}_1, \vec{q}_2)$, we need to perform a change of basis, i.e., write the Bresenham circle as a linear combination of the basis $(\vec{q}_1, \vec{q}_2)$. In other words, we sample the texture map at locations

$$(x_k, y_k) = (u_k\xi_1 + v_k\xi_2, u_k\eta_1 + v_k\eta_2), \quad k = 1, ..., 16. \tag{4}$$

The corner test is then performed on the obtained samples. Some of these samples might be unnecessary for the corner test: AGAST allows to reduce such needless sampling operations and save time.

The idea of performing AGAST in local axes is illustrated in Fig. 4. Non-local maxima suppression is then applied on the generated score map in order to select the keypoint candidates.

Multiscale detection. For improved stability to significant scale changes we run AGAST test on each level of a multiscale image pyramid. The pyramid consists of the original image and its subsampled versions (*octaves*); each next level is halfsampled with respect to the previous level. After the keypoint is detected on a given level, it is kept only if its AGAST score is greater than AGAST scores in the same position in an adjacent level. Differently to the original BRISK, the pyramid we use is sparse, i.e., there is only one level per octave. This is mainly motivated by the fact that we do not use the pyramid to derive the keypoint scale, but need it only to avoid missing keypoints when the image scale changes significantly.

Keypoint scale selection. To derive the keypoint scale we exploit the depth map similarly to [49]. A typical corner revealed by AGAST is an intersection of two straight contours or a point-like structure. We believe that the characteristic size of such a structure (its *visual scale*) is difficult to define properly: local patches of slightly different sizes centered around such a corner are visually similar, contrarily, for example, to a blob-like structure which exhibits more clearly such a characteristic size. However, scale estimation accuracy has a major impact on repeatability. For this reason, we use AGAST response only to derive the keypoint position but not its scale, since in case of RGBD images a better clue of scale is available in the depth map. To achieve scale invariance, we employ the *geometrical* scale. Namely, we get the keypoint scale from the depth map assuming that the underlying visual detail is of a fixed spatial size σ_0. As observed also in [49], the geometric scale is inversely proportional to depth, i.e., keypoints farther from the camera have smaller spatial support in pixel units,

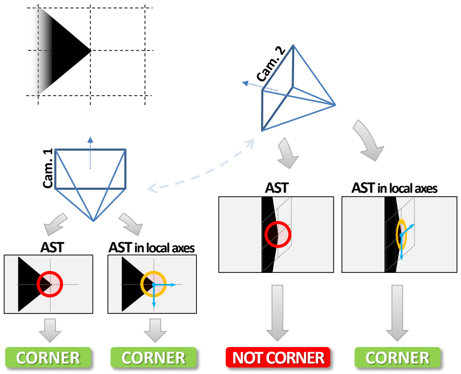

Figure 4: Illustration of application of Accelerated Segment Test (AST) in standard image axis versus local axes derived from the depth map. A corner viewed under a large angle projects itself at a nearly straight contour on the camera plane, so that the corner test in standard image axes fails causing a repeatability loss.

due to perspective distortion. σ_0 is the coefficient of this inverse proportionality relation. It defines a sort of "anchor" size to which objects (in spatial units measured in the camera plane) are scaled based on their depth. Intuitively, σ_0 is related to the characteristic size of repeatable landmarks, which depends on the content and viewing conditions. The optimal value of σ_0 is found by grid search as explained in Section 4.4. Hence, the resulting scale is simply equal to $\sigma = \frac{\sigma_0}{z}$, where z is the average depth of the keypoint. This gives a rough initial scale estimation which is then refined on the descriptor stage: to avoid scale estimation errors for keypoints situated near depth boundaries, we estimate z iteratively, at the same time when the keypoint dominant orientation is selected. This is explained further in Section 3.3. The keypoint area is finally described by an ellipse spanning the scaled local axes $\sigma\vec{q}_1$ and $\sigma\vec{q}_2$. Thus, TRISK keypoints are not circular as those of SIFT or BRISK, but elliptical similarly to the keypoints produced by affine-covariant detectors [33].

3.2.3. Local maxima filtering

The initial keypoint candidates given by local maxima of AGAST score are then analyzed subject to their stability. A well-known supplementary criterion to filter out unstable keypoint candidates is based on Harris cornerness measure [58]. It was first used in SIFT and then reemployed in other detectors, e.g. ORB [25]. Some keypoints reported by a corner detector may actually be situated on straight edges, for example due to aliasing artifacts. These keypoints are prone to localization errors. In order to filter them out, the eigenvalue ratio of Hessian matrix H is thresholded [14]:

$$H = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}. \tag{5}$$

Here I denotes the smoothed texture image.

In our approach, differently to the presented classic technique, we replace the standard derivatives of I by the directional derivatives computed in the adaptive local axes $\vec{q}_1$ and $\vec{q}_2$, i.e. we deal with the eigenvalues of

$$H_q = \begin{pmatrix} I_{\vec{q}_1\vec{q}_1} & I_{\vec{q}_1\vec{q}_2} \\ I_{\vec{q}_1\vec{q}_2} & I_{\vec{q}_2\vec{q}_2} \end{pmatrix}. \tag{6}$$

The reason is always the same: changing the axes allows to reduce the impact of perspective distortions when dealing with the texture curvature. We compute the eigenvalue ratio in the same way as in SIFT, and use the same threshold value: a keypoint is rejected if the ratio is greater that 10 [14].

3.2.4. Accurate localization

On the last stage of the detection process, we perform an accurate localization of the remaining keypoint candidates. This allows to localize accurately the keypoints detected on subsampled versions of the input image and also serves as an additional criterion of keypoint stability: not all the keypoint candidates may be precisely localized, and the ones that reveal unstable behavior during the accurate localization are rejected.

We reemploy the interpolation technique used in SIFT and SURF and initially presented in [59], based on the Taylor expansion of the score function up to the quadratic terms. We apply it to the AGAST score reducing the number of dimensions from three to two, as no scale dimension is considered in our case, and in the adaptive local axes instead of the standard ones.

More precisely, let S be the AGAST score, (x, y) a candidate point, (x^*, y^*) an accurately localized local maximum, and $Q = (\vec{q}_1\ \vec{q}_2)$ the coordinate transformation. We first express S in the local coordinates:

$$\tilde{S}(\xi, \eta) = S\left(Q\begin{pmatrix}\xi\\ \eta\end{pmatrix} - \begin{pmatrix}x\\ y\end{pmatrix}\right). \tag{7}$$

We develop the Taylor expansion of $\tilde{S}(\xi^*, \eta^*)$ where $(\xi^*\ \eta^*)^T = \vec{\delta} = Q^{-1}\begin{pmatrix}x^* - x\\ y^* - y\end{pmatrix}$ with respect to the local coordinate center:

$$\tilde{S}(\xi^*, \eta^*) \approx \tilde{S} + \left(\tilde{S}_\xi\ \tilde{S}_\eta\right)\vec{\delta} + \frac{1}{2}\vec{\delta}^T\begin{pmatrix}\tilde{S}_{\xi\xi}\ \tilde{S}_{\xi\eta}\\ \tilde{S}_{\xi\eta}\ \tilde{S}_{\eta\eta}\end{pmatrix}\vec{\delta}. \tag{8}$$

$\tilde{S}$ and its derivatives on the right side of the equation above are taken at point $(0, 0)$. Deriving this and using the fact that (ξ^*, η^*) is a local maximum, i.e., $\tilde{S}_\xi\Big|_{\xi^*,\eta^*} = \tilde{S}_\eta\Big|_{\xi^*,\eta^*} = 0$, we obtain:

$$\vec{\delta} = -\begin{pmatrix}\tilde{S}_\xi\\ \tilde{S}_\eta\end{pmatrix}\begin{pmatrix}\tilde{S}_{\xi\xi} & \tilde{S}_{\xi\eta}\\ \tilde{S}_{\xi\eta} & \tilde{S}_{\eta\eta}\end{pmatrix}^{-1}. \tag{9}$$

The displacement in standard image axes is equal to $Q(\vec{\delta})$.

Similarly to the SIFT implementation [60] we apply this process iteratively, cumulating the offset and reinterpolating the derivatives of $\tilde{S}$. For a better selection of stable keypoints, we reject a keypoint during the iterations if the Hessian of $\tilde{S}$ is rank-deficient. Following [60], in our implementation we perform at most 5 iterations.

3.3. The Descriptor

Once the set of interesting point positions and scales is provided, a compact description is computed for each point.

In our previous work [52], we studied how binary features may be used to extract a surface-intrinsic information from RGBD images in order to provide a description robust to rigid 3D deformations. A descriptor sampling pattern was projected on the scene surface, providing a depth-based descriptor normalization procedure aimed at producing invariant features. However, such a projection is (1) very sensitive to depth map noise and (2) requires a high computational effort. To be robust to the viewpoint position changes on the descriptor level, in this work we propose a simpler approach based on a similar concept: the descriptor normalization is performed according to the local tangent plane approximating the scene geometry nearby the keypoint, computed directly in the camera coordinates using the definition of local axes in Section 3.2.1.

Non-binary local planar normalization-based descriptors are studied in the literature [37, 38, 50, 51]. In this work we apply this principle to produce a binary descriptor. Precisely, we reuse the BRISK descriptor sampling pattern, applying it to the image in adaptive local axes computed at the keypoint that immediately gives us the approximating local plane. The pattern used in the original BRISK implementation and an example of how it is mapped onto the scene using local axes at a given corner point is shown in Fig. 5. We notice that our design is not restricted to the BRISK sampling pattern; another manually designed or appropriately learned pattern, e.g. [27] or [25], might be used with no additional cost.

In TRISK we proceed as follows. Let $\{(\upsilon_k, \nu_k)\}_{k=1}^{M}$ represent the Cartesian coordinate pairs of the descriptor sampling pattern points. In case of BRISK, $M = 60$. As discussed in [52], (υ_k, ν_k) values may be easily derived analytically thanks to the radially regular disposition of the pattern points.

For a given keypoint position (X, Y) and scale σ, we reuse the local axes $\vec{q}_1$ and $\vec{q}_2$ in order to map the pattern points to the image plane, similarly to the detector pattern sampling in Eq. (4):

$$\begin{pmatrix} x_k \\ y_k \end{pmatrix} = \sigma \upsilon_k \vec{q}_1 + \sigma \nu_k \vec{q}_2 + \begin{pmatrix} X \\ Y \end{pmatrix} \tag{10}$$

Notice that, differently from (4), here we use a different pattern, indicated by $\{(\upsilon_k, \nu_k)\}_{k=1}^{M}$, which is scaled by σ, while in (4) the spatial extent of the pattern was fixed. The original BRISK uses a two-pass scheme that consists in sampling the pattern, computing its dominant orientation from obtained samples and sampling the oriented version of the pattern (by a “pass” we mean sampling the pattern). In TRISK we proceed similarly. However, the descriptor pattern in our case is more sensitive to keypoint parameter estimation errors due to (a) perspective warping introduced by the local axes, (b) depth map imperfections and (c) scale errors for keypoints situated near object boundaries, where the depth varies abruptly. The latter is crucial since we average depth to derive the geometric keypoint scale as explained above. For this reason, we propose the following three-pass scheme that estimates accurately both the dominant orientation and scale.

We begin with the geometric scale $\sigma = \frac{\sigma_0}{z}$, where z is an average depth value in the keypoint center. This provides a rough initial estimate of the scale which is further refined.

1. The pattern is sampled in locations (x_k, y_k): averaged image intensity is computed at each point. The neighborhood radius per point is taken as shown in Fig. 5 and scaled by σ. The pattern is sampled both from texture and depth maps, producing two sets of smoothed intensity and depth values P_I and P_D, respectively.

2. The descriptor dominant orientation Θ is computed using the BRISK methodology from P_I; the depth value z used in the initial estimate of the scale is recomputed as average of all the values of P_D.

3. The unmapped pattern (υ_k, ν_k) is reoriented according to Θ: each point is simply turned around the pattern center by $-\Theta$ radians. The new oriented pattern (υ'_k, ν'_k) is used, together with the updated value of σ, to sample the texture by applying Eq. (4).

4. Dominant orientation Θ and scale σ are re-estimated once again in the same way as in step 2, producing final values Θ^* and σ^*.

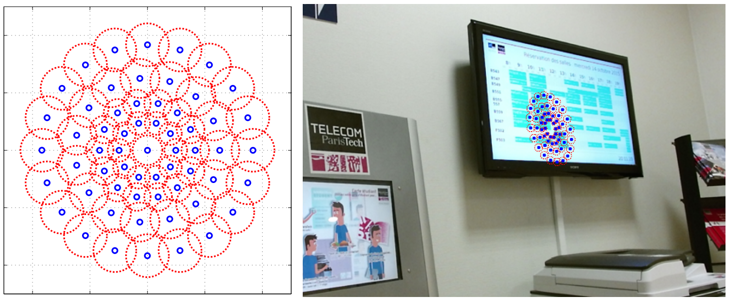

Figure 5: BRISK descriptor sampling pattern from the original implementation (left) and its mapping to the surface through local planar normalization (right).

5. The pattern is sampled again according to Θ^* and σ^*, giving final P_I and P_D sets.

6. Control scale value σ_c is computed as before; the keypoint is kept only if σ_c differs from σ^* by no more than 1% of the latter, i.e., if the scale error is negligible.

7. Finally sampled P_I values undergo pairwise intensity comparison tests to produce a binary string forming the descriptor.

The descriptor interoperability between cameras with different intrinsic parameters is achieved by a proper choice of σ_0. If σ_0^* is a reference value for Kinect expressed in the same units as the depth (e.g., the one we obtain in Section 4.4), W^* and ω^* are its image width in pixels and its horizontal angle of view, respectively, the interoperability with another sensor having intrinsic parameters (W, ω) is ensured if

$$\sigma_0 = \frac{W}{W^*} \frac{\tan \frac{\omega^*}{2}}{\tan \frac{\omega}{2}} \sigma_0^*. \tag{11}$$

This is derived using the pinhole camera model and assuming that σ_0 corresponds to a fixed spatial size [52].

3.4. Implementation details

TRISK has several parameters that control different stages of the feature extraction process. For most of them we use the same values as in the original

BRISK or SIFT papers or their implementations [14, 26, 60]. Other parameters, such as the 1%-error threshold in the scale estimation, are derived from experiments and do not impact significantly the performance. All these values are mentioned in the text. The remaining parameters are: (1) neighborhood size factor κ for PCA-based normal estimation used when computing the adaptive local axes, (2) AGAST score threshold t and (3) basic scale σ_0 used in the scale selection. A discussion of their appropriate values based on the matching performance is given in Section 4.4.

For all the texture smoothing and interpolation operations we use the image filter presented in [61].

The depth map values are used for normal estimation and scale selection. In both cases, they are not used directly, but a neighborhood of each pixel is considered. This allows to cope with the noise and small "holes" (areas with no depth). Larger "holes" are simply skipped (i.e., no keypoint detection is performed in these areas).

4. Experiments

In this section, we evaluate the proposed method compared to several well-known local visual features in two scenarios:

- a mid-level feature evaluation in terms of matching score and receiver operating characteristics (ROC) similarly to [6, 38, 52, 54, 56] performed on synthetic RGBD data and RGBD images from the *Freiburg* dataset [62] acquired with a Microsoft Kinect sensor;
- a visual odometry experiment on three sequences of the *Freiburg* dataset.

In the following, we provide a detailed description of the experiments and discuss the results.

4.1. Compared methods

The following local feature extraction methods are used in the experiments.

- The baseline is given by the BRISK features [26], computed on the RGB channels only (ignoring depth). The publicly available original implementation is used.

- BRAND descriptor [49] is a recent approach for RGBD content matching. We use it in conjunction with STAR detector as proposed in the original paper. This method is referred to as STAR-BRAND. STAR is an OpenCV implementation of the Center Surround Extrema (*CenSurE*) [63]. The original implementation of the descriptor is used.

- VIP [37] is based on SIFT descriptors computed on RGBD images and aimed at improved viewpoint invariance. We use publicly available authors implementation.

- As we deal with out-of-plane rotations, we compare the proposed method to an affine-covariant detector [33] initialized with SIFT keypoints and referred to as AFFINE. *VLFeat* [60] implementation is used.

- For completeness, standard SIFT features [14] computed on RGB channels only are also involved in the evaluation (*VLFeat* implementation is used).

We hence have six approaches being compared. Table 1 summarizes some characteristics of the compared methods.

4.2. Datasets

We measure the performance of TRISK on several synthetic RGBD sequences [2] we used in our previous works [52, 54, 56] and three RGBD sequences from *Freiburg* dataset [62]. The images are obtained using static 3D scenes, rendered from different viewpoints. The scene content is mainly composed of several publicly available textured 3D models[3] with various texture and geom-

[2] The dataset is available for download at the address `http://webpages.l2s.centralesupelec.fr/perso/giuseppe.valenzise/download.htm`

[3] 3D model courtesy of `http://archive3d.net` and `http://www.turbosquid.com`, accessed in Oct.-Nov. 2013

Method	Keypoint type	Descriptor type and size	Depth map use
TRISK	Corner	Binary 512 bit	detector and descriptor
BRISK	Corner	Binary 512 bit	no
STAR-BRAND	Blob	Binary 512 bit	descriptor
VIP	Blob	Numeric 128 dim.	preprocessing
AFFINE	Blob	Numeric 128 dim.	no
SIFT	Blob	Numeric 128 dim.	no

Table 1: Summary of compared methods.

etry characteristics. The *Graffiti* sequence is synthesized from the frontal view of the original Graffiti sequence [6]. Being synthetically generated, this dataset provides a highly accurate ground truth for the mid-level feature evaluation. As we are mainly interested to the invariance to viewpoint position changes, all the sequences contain significant changes in camera position between views (examples of images are shown in Fig. 6):

- *Bricks*: 20 images with large out-of-plane rotations (up to 90°) and vertical camera displacements,

- *Graffiti*: 25 images with yet larger out-of-plane rotations (up to 120°); this RGBD sequence is resynthesized from the frontal image of the original *Graffiti* sequence [6],

- *House*: 25 images captured with a camera flying back, giving significant scale changes and limited out-of plane rotations (up to 25°).

The *Freiburg* dataset [62] consists of several indoor RGBD image sequences of 640×480 pixels acquired with Microsoft Kinect and ASUS Xtion sensors. Ground truth sensor position and orientation is tracked using a motion-capture system, making this dataset suitable for SLAM and visual odometry experiments. The depth maps are of a standard Kinect quality (may contain regions

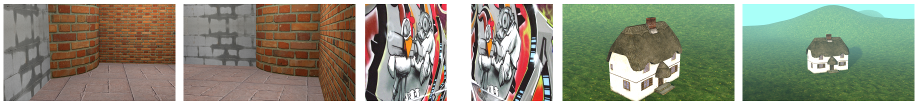

Figure 6: Texture maps of first and last view of *Bricks*, *Graffiti* and *House* RGBD sequences (from left to right) used in the matching score and ROC tests.

with undefined depth). *Freiburg* sequences contain more complex camera position changes. The sequences *desk* (we used 40 frames with 10 frames skipping) and *structure_texture_far* (59 frames with 5 frames skipping) represent out-of-plane rotations, whereas in the *floor* sequence (19 frames with 5 frames skipping) the camera moves arbitrarily within the scene.

4.3. Matching score and ROC

We first test the matching capabilities and the discriminability of the proposed features following the protocol initially established by Mikolajczyk et al. in [5, 6]. In different variants, this kind of evaluation frequently appears in the literature (e.g. [4, 7, 8, 9, 11]), and has become classic for mid-level evaluation of local image features.

In this section, we first revisit the evaluation framework taking into account the extended modality (presence of “D” in “RGBD”). The test setting is resumed in the following steps.

I. A set of RGBD image sequences is taken with each sequence representing a certain class of visual distortions.

II. In each sequence, its first image is taken as the reference and matched against each remaining image. The reference descriptors are further referred to as *matchees*, whereas the test descriptors are called *matchers*. The matching consists in finding the closest matcher to each matchee. The inter-descriptor similarity measure (*score*) depends on the descriptors type. Hamming distance, i.e., number of bit positions where matcher and matchee take different values, is used for all the binary descriptors. As explained in [14], the ratio $\rho_{1/2}$ of Euclidean distances “matchee – closest matcher” and “matchee – 2nd closest matcher” is used for SIFT-based descriptors; this similarity measure gives a

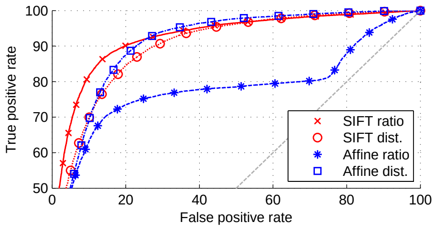

Figure 7: SIFT descriptor matching using different inter-descriptor similarity measures. Simple distance-based matching is compared to $\rho_{1/2}$ ratio-based matching [14] for standard (blue) and affine normalized (red) SIFT descriptors. To plot ROC, 20K true positive and 20K false positive matches were collected by matching the test sequences in Fig. 6. Normal SIFT descriptors are more distinctive when being matched using ratio-based score, whereas affine invariant features perform much better with simple Euclidean distance. The best performing scores are used in further tests in this paper.

significant discriminability gain with respect to the simple Euclidean distance between the descriptors. However, we employ the simple Euclidean distance for SIFT descriptors issued from affine covariant keypoints as, in this case, the above mentioned ratio causes the losses of distinctiveness, as we discovered previously in [38]. This choice of scores is also validated experimentally on the data we use as presented in Fig. 7.

III. The set of matching feature pairs between the two given images (*putative matches*) is split into correct (*true positive*) and incorrect (*false positive*) matches using ground truth. Two keypoints coming from different images but occupying the same area of the scene are called *repeated keypoints*; they produce a correct match if the descriptors corresponding to these keypoints are matched. The keypoint area overlap is controlled by means of the *overlap error*:

$$\epsilon(A, B) = 1 - \frac{A \cap B}{A \cup B}. \tag{12}$$

A positive match is then labeled as "true" if $\epsilon(A, B) < \epsilon_0$, where ϵ_0 is typically equal to 0.5. Originally, A and B were representing the elliptical keypoint regions projected on the same camera plane (for example, the reference one) [6]. Thus, ϵ represented the degree of overlapping of two "spots" each highlighting a keypoint. However, if the observed scene is not entirely planar, the reprojected "spots" are not elliptical and may take arbitrary not even connected shapes. Their overlap then can not be computed analytically. For this reason, here we follow our previous works [52, 54, 55, 56] and consider the overlap of 3D spheres centered at keypoint positions projected on the scene surface. The radius is selected in such a way that the keypoint ellipse may be backprojected from the camera plane onto a 3D circle that fits the sphere boundary. As the camera positions and orientation matrices are provided, the necessary pixel-level ground truth is derived by depth maps backprojections. In our tests, each matchee may match at most one true positive matcher (we take the one that minimizes $\epsilon(A, B)$).

IV. The ratio between the number of correct matches and the maximum possible number of matches is reported as *matching score* per image pair.

V. A putative match is found if the matching distance between two descriptors is below a certain threshold. By varying the value of this threshold, one can compute the true and false positive rates and trace the ROC curve. The ROC curves are balanced, i.e., an equal number of matching pairs of each class (true and false) is randomly selected among all the matches issued from each scene.

Matching score allows to judge on the ability of the detector to produce repeatable keypoints as well as on the matching capability of the entire pipeline, whereas ROC shows how the descriptors are discriminative, e.g., their ability of distinguishing salient visual information in presence of deformations. Put together, these characteristics trace the two main axes of the local visual features mid-level evaluation: *repeatability* and *distinctiveness*.

The resulting matching score and ROC curves obtained on the test sequences are presented in Fig. 8 and 9. The number of features detected by each method is reported in Table 2. It can be seen from the results that in all the test sequences TRISK demonstrates improved overall matching score. In some cases (*Graffiti*, *House*, *Floor*) TRISK also shows the slowest decay, which indicates improved feature stability under viewpoint position changes. The second best matching score on synthetic sequences (top row in Fig. 8) is arguably achieved by VIP. Based on a planar normalization technique, VIP performs well in case of simple geometry, i.e., when the scene surface is mostly planar or very smooth, otherwise it may even be unable to detect any features. TRISK also exploits the principle of planar normalization, but in a much more local way, which allows it to perform well in scenes with more complex geometry, such as *desk* and *House*.

As for the descriptor discriminability examined with ROC curves (bottom rows in Fig. 8 and 9), the best performance is shared among TRISK, VIP and sometimes SIFT. TRISK outperforms the other approaches on sequences with simple geometry and detailed texture (*Graffiti* and *structure_texture_far*), but in other cases turns out to be comparable to or moderately less distinctive than non-binary descriptors, notably SIFT and VIP. This result deserves a more elaborated discussion.

First, the non-binary descriptors in our tests are represented by 128-dimensional numeric vectors. They are naturally more distinctive than the 512-bit binary descriptors since they carry more information. This is coherent to other evaluations in the literature [8, 26, 9]. It is also worth noticing that the other binary competitors are mostly always singificantly outperformed by TRISK.

Second, the observed moderate ROC gains of non-binary features over TRISK is arguably meaningful. In the *House* sequence VIP demonstrates the best discriminability but low matching scores: only the first 8 views are reliably matched against the reference. Consequently, the majority of the true positive matches comes from these views. However, the first views have less perspective distortions compared to the reference than the others, and thus the matched descriptors from the first views are less deformed, and their corresponding true and false positives are easier to distinguish by the inter-descriptor difference. This leads to a gain in terms of ROC, whereas the most challenging part of the sequence remains mostly unmatched. Hence, ROC is comparable only if the matching score is reasonably high over the whole deformation spectrum. Even though to a lesser extent, the other sequences exhibit a similar phenomenon.

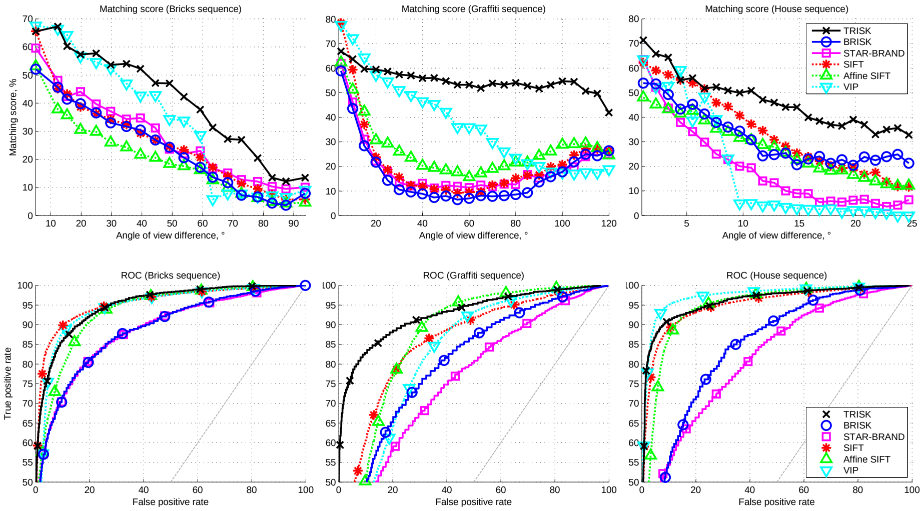

Figure 8: Matching score and receiver operating characteristics demonstrating repeatability and distinctiveness of the compared detectors and descriptors, mainly under out-of-plane rotations (*Bricks* and *Floor* sequences) and scale changes (*House* sequence). Computed on synthetic RGB data. At least 4800 true positive and 4800 false positive matches were selected to plot each ROC curve.

4.4. Parameter values estimation

The matching score and ROC are also used to find empirically optimal values for TRISK parameters. To do this, we collected 500 image pairs from *large_with_loop* and *long_office_household* Freiburg sequences, respectively. These two sequences represent different kinds of viewpoint position changes (from out-of-plane rotations in *long_office_household* to scale changes and 3D translations in *large_with_loop*). We consider the following space for the grid search: we take 6 values of support size factor κ used in the normal estimation, 6 values of basic scale σ_0 and 5 values of AGAST score threshold t. This gives in total $6{\times}6{\times}5$ triples $(\kappa_i, \sigma_{0i}, t_i)$, that cover a spectrum of reasonable values for the input parameters. We matched then all the selected image pairs using each parameter triple. This provided us with about 20 millions matching pairs of features in total. As a function $\mathcal{F}$ to maximize, we choose the product of averaged matching score over all the image pairs and area under ROC curve, which seems a

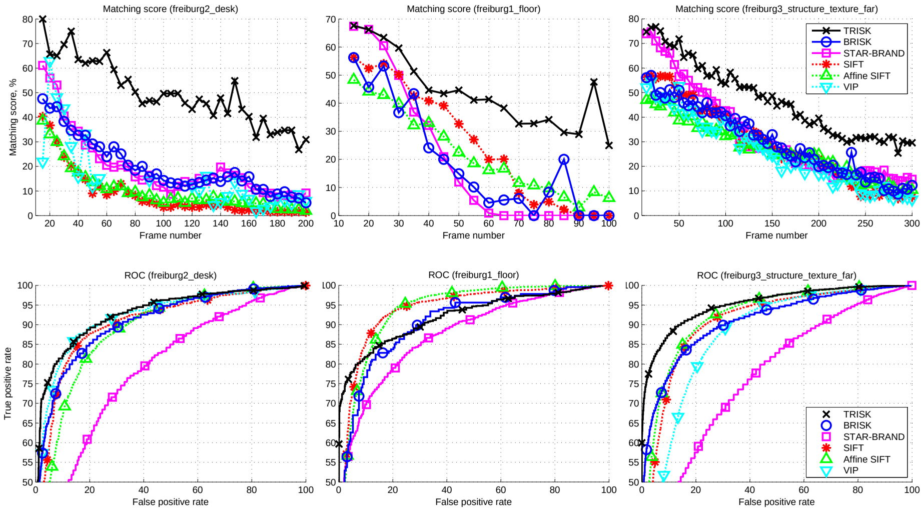

Figure 9: Matching score and receiver operating characteristics demonstrating repeatability and distinctiveness of the compared detectors and descriptors under viewpoint position changes of different kind. Computed on three sequences of *Freiburg* dataset [62], acquired with Kinect. In some images in *desk* sequence and in the whole *floor* sequence VIP turns unable to detect any feature.

reasonable joint performance index of detector and descriptor.

We notice that the AGAST score threshold t in TRISK plays the same role as in BRISK, and has a major impact on the number of detected features. However, when it varies in a reasonable range, it does not produce a significant impact on the performance index: when averaging $\mathcal{F}$ over t_i, the standard deviation in the (κ, σ_0) plane does not exceed 0.014 when $\mathcal{F}$ varies in the range 0.12 to 0.22. Based on this, we averaged $\mathcal{F}$ over 5 parameters of t, reducing the search space to two dimensions (κ, σ_0), where $\mathcal{F}$ exhibits a distinctive maximum near point $(\kappa^*, \sigma_0^*) = (\mathbf{25}, \mathbf{14.27})$. The contour plot of $\mathcal{F}$ in Fig. 10 allows further analysis: when σ_0 is large enough, (i) the performance depends mainly on κ, (ii) it does not vary significantly after κ becomes reasonably high. This result is coherent, since κ is introduced to cope with the depth map noise, and is rather a depth sensor characteristic, while σ_0 may be content-dependent, as it reflects a characteristic size of repeatable landmarks observed in the training data. Consequently, we

Sequence		TRISK	BRISK	BRAND	SIFT	AFFINE	VIP
Bricks	min	493	766	1072	1638	2194	3346
	avg	1329	915	1188	1841	2482	4293
	max	1840	1163	1330	2047	2714	5458
Graffiti	min	994	855	595	782	1079	1603
	avg	1518	1041	809	1305	1764	2280
	max	1804	1151	917	1615	2171	3029
House	min	393	164	462	1924	2445	237
	avg	879	231	889	2235	3056	1831
	max	1302	276	1240	2637	3609	3503
desk	min	111	194	433	898	1115	0
	avg	214	421	524	1036	1343	113
	max	296	689	611	1213	1597	420
floor	min	311	32	431	1049	1328	0
	avg	578	172	700	1257	1634	2
	max	777	357	1045	1460	1895	59
structure_texture_far	min	579	156	509	1060	1461	672
	avg	913	471	692	1154	1615	976
	max	1210	732	838	1298	1820	1220

Table 2: Minimal, average and maximal number of features extracted from each scene. Minimum and maximum values per row are highlighted in green and yellow.

recommend the found values (κ^*, σ_0^*) as default for Kinect depth maps given in meters, and use in all the experiments in this paper, except the ones on the synthetic dataset (Fig. 8). In this case, the depth maps are quasi-perfect (contain no noise), thus a small value of κ is more appropriate (we used $\kappa = 5$). As for σ_0, even if it requires a proper tuning, as the observed content might be rather different from the Kinect one, we simply rescaled the depth values to fit Kinect statistics and use the same value of $\sigma_0 = \sigma_0^*$.

4.5. Visual odometry

In addition to the mid-level evaluation, we assess TRISK performance in a visual odometry scenario using two Kinect and Asus Xtion image sequences from *Freiburg* dataset [62]. The goal consists in retrieving camera pose evolu-

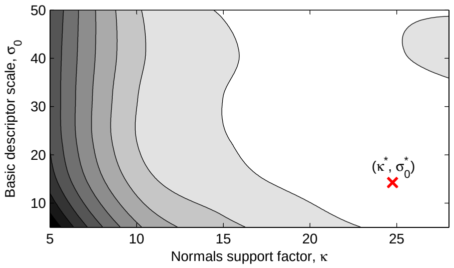

Figure 10: Contour plot of the performance index $\mathcal{F}$ in the plane (κ, σ_0).

tion relatively to an initial pose using only the acquired images. The ground truth pose is recorded with a motion capture system and is provided within the dataset. We follow the setting of [49]: to compute the camera transformation (translation and rotation) between two frames, we match them, apply RANSAC to filter putative matches and, finally, run the Iterative Closest Point algorithm [64] retrieving the relative translation vector and rotation matrix. The resulting pose is recovered by cumulating deduced translations and rotations. In this experiment we limit the number of keypoints extracted from each image by each detector, keeping at most 1000 keypoints with the highest response. In case of TRISK, the detector response is the interpolated AGAST score.

Two types of errors are used in the evaluation:

- *translation error*: the distance between estimated and ground truth positions,
- *rotation error*: $\varepsilon = \arccos \frac{\mathrm{tr}(R^{-1}R_{gt})-1}{2}$, where R is the estimated camera orientation matrix with respect to the initial pose, and R_{gt} is the ground truth one.

Typically, each registered frame is matched against the next one, providing a "delta-pose" that is added to the current position. In our experiment, we proceed differently: we skip more than one frame, i.e., we look for the transformation relating frame 0 to frame $K > 1$, then frame K to frame $2K$, etc. This technique has a twofold effect. On one hand, it allows to compensate the visual drift being cumulated with each new "delta", as well as to reduce the

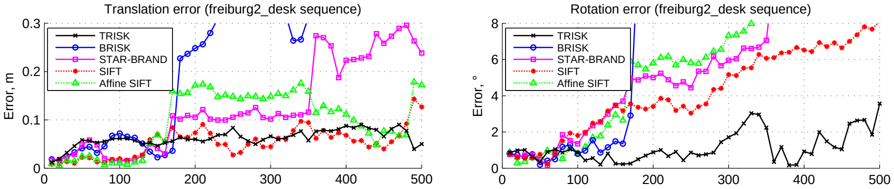

Figure 11: Visual odometry with 10 frames skipping on *freiburg2_desk* sequence (first 500 frames): translation (top) and rotation (bottom) errors. VIP fails on this sequence, thus it is not reported.

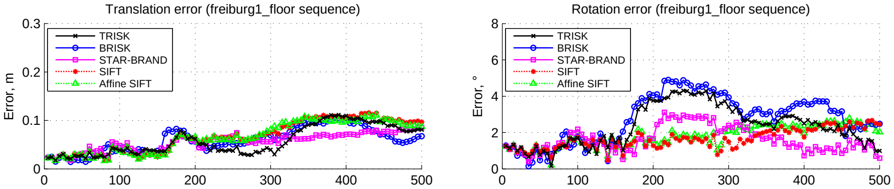

Figure 12: Visual odometry with 5 frames skipping on *freiburg1_floor* sequence (first 500 frames): translation (top) and rotation (bottom) errors. VIP fails on this sequence, thus it is not reported.

computational time. On the other hand, the resulting errors depend strongly on the features quality (matching capabilities and localization accuracy), as the visual difference between frames n and $n + K$ is typically more significant than the one between n and $n + 1$. This setting is thus a good scenario to evaluate the features.

Translation and rotation errors evolution on different sequences is presented in Fig. 11, 12 and 13. To compensate for the randomness induced by RANSAC, we run the experiment 10 times on each sequence and then averaged the results.

All the methods have similar error values in the first frames. However, as the scene evolves, the drift cumulates differently for different features. It can be observed that TRISK generally achieves smaller errors. An exception is the *floor* sequence (Fig. 12), where all the methods achieve small errors compared to other sequences (less than 12 cm and 5°), but AGAST-based features turn out to be slightly less precise in rotations. The possible reason is that in this sequence the camera moves quickly (for this reason we set $K = 5$ for this sequence and

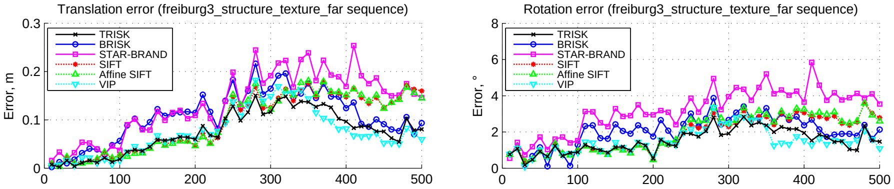

Figure 13: Visual odometry with 10 frames skipping on *freiburg3_structure_texture_far* sequence (first 500 frames): translation (top) and rotation (bottom) errors.

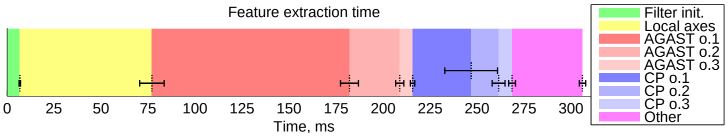

Figure 14: Feature extraction time averaged over images from matching score test (Fig. 9). Smoothing filter initialization, local axes computation, AGAST over 3 octaves, keypoint candidates processing ("CP") over 3 octaves (includes accurate localization, Harris corner test and descriptor computation) and remaining processing times and their standard deviations are displayed.

not 10 as for the others). This causes a noticeable directional blur in texture maps, which interferes with corner detection but is manageable by blob detectors. A drastic difference in the odometry precision is revealed on *desk* sequence (Fig. 11), where mostly all the other approaches, notably BRISK, experience severe errors in matching consecutive frames. TRISK is the only approach providing precision within 10 cm and 4°. Finally, on *structure_texture_far* sequence (Fig. 13), TRISK is mainly competing with VIP, which also performs well thanks to the locally planar geometry. It is worth noticing that on the other two sequences VIP proves unable to provide enough matches for continuous trajectory estimation.

4.6. Note on computational efficiency

We ran our tests on a 64-bit Windows machine with a 3.5 GHz 6-physical core CPU and 16 Gb of RAM. Figure 14 reports the time spent on each stage of feature extraction from real RGBD images in the *Freiburg* dataset. Being invoked from MATLAB environment through MATLAB MEX interface, our

C++ TRISK implementation takes **306 ms** per VGA image (average over about 150 images, with 21.2 ms standard deviation). This corresponds to about 540 μs per feature. The most time consuming steps are the local axes computation and AGAST on the first octave. The description time is included in the keypoint candidates processing on each octave, and thus is much lower than the detection time.

Compared to other conventional RGB binary features, TRISK entails a higher computational cost: it is about 8 times slower than FREAK [27] and more than 20 times slower than BRISK [26] and ORB [25]. This overhead is certainly due to the fact that TRISK processes also the geometric information, in particular, by computing per pixel local axes, as well as to the fact that our implementation might be further optimized. As we noticed before, the local adaptive axes might be computed differently, e.g., PCA-based normal estimation technique [57] may also provide two orthogonal vectors to the normal that might be used as the local axes. This, however, requires the complete PCA decomposition of the point cloud covariance matrix at each pixel. We tested this approach and obtained very similar performance, but the average local axes computation time increased by 60 ms.

TRISK can be speeded up considerably by using multiple threads. The adaptive local axes computation, AGAST and local maxima suppression are purely local, and all the keypoint candidates are processed independently starting from the accurate localization to the descriptor computation. This makes TRISK easily parallelizable, allowing for distributed and GPU-based implementations.

5. Conclusion and future work

In this paper we presented a complete pipeline of local feature extraction for texture-plus-depth image matching. The proposed TRISK features target application scenarios where significant viewpoint position changes are present in the input data. The experiments showed that TRISK improves consistently both feature stability and distinctiveness, which allows for better performance on the application level. TRISK can be applied on real RGBD images acquired with low-cost RGB-depth camera pair, such as Microsoft Kinect or Asus Xtion, without any complex preprocessing of the depth map. The computational effort required to process an image is sufficiently low, so that it is able to perform at near-realtime rates. A publicly available implementation of TRISK can be downloaded at the address `http://webpages.l2s.centralesupelec.fr/perso/giuseppe.valenzise/download.htm`.

Clearly, TRISK could be improved, notably in its ability to deal with complex, highly detailed geometry, currently limited by the local planar approximation used to compute the descriptor. A more complex way to render the descriptor stable and invariant to viewpoint position changes, such as [52], is more computationally expensive and sensible to the depth map imperfections. Rendering the descriptor robust to geometrically complex scenes is one of the main objectives for our future work. Along with this, a learning-based descriptor design [25, 29] seems promising from the discriminability boosting point of view.

[1] A. Gil, O. M. Mozos, M. Ballesta, O. Reinoso, A comparative evaluation of interest point detectors and local descriptors for visual SLAM, Machine Vision and Applications 21 (6) (2010) 905–920.

[2] B. Kitt, A. Geiger, H. Lategahn, Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme, in: Proceed. of IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 2010.

[3] S. N. Sinha, J.-M. Frahm, M. Pollefeys, Y. Genc, Feature tracking and matching in video using programmable graphics hardware, Machine Vision and Applications 22 (1) (2011) 207–217.

[4] F. Fraundorfer, H. Bischof, A novel performance evaluation method of local detectors on non-planar scenes, in: Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec., San Diego, USA, 2005.

[5] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Machine Intell. 27 (10) (2005) 1615–1630.

[6] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, A comparison of affine region detectors, Intern. J. of Comp. Vision 65 (1-2) (2005) 43–72.

[7] P. Moreels, P. Perona, Evaluation of features detectors and descriptors based on 3D objects, Intern. J. of Comp. Vision 73 (3) (2007) 263–284.

[8] J. Heinly, E. Dunn, J.-M. Frahm, Comparative evaluation of binary features, in: Proceed. of Europ. Conf. on Comp. Vision, Springer, Firenze, Italy, 2012.

[9] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, R. Cilla, Evaluation of low-complexity visual feature detectors and descriptors, in: Proceed. of IEEE Intern. Conf. on Dig. Signal Proc., Fira, Santorini, Greece, 2013.

[10] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, N. M. Kwok, A comprehensive performance evaluation of 3d local feature descriptors, Intern. J. of Comp. Vision (2015) 1–24.

[11] D. Mukherjee, Q. J. Wu, G. Wang, A comparative experimental study of image feature detectors and descriptors, Machine Vision and Applications 26 (4) (2015) 443–466.

[12] ISO/IEC JTC 1/SC 29/ WG 11, ISO/IEC CD 15938-13 compact descriptors for visual search, MPEG document N14681, ISO/IEC, Sapporo, Japan (July 2014).

[13] ISO/IEC JTC 1/SC 29/ WG 11, CDVA: Requirements, MPEG document N14509, ISO/IEC, Valencia, Spain (March 2014).

[14] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Intern. J. of Comp. Vision 60 (2) (2004) 91–110.

[15] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, Speeded-up robust features (SURF), Comp. Vision and Image Understanding 110 (3) (2008) 346–359.

[16] V. Balntas, L. Tang, K. Mikolajczyk, Binary online learned descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[17] Y. Duan, J. Lu, J. Feng, J. Zhou, Context-aware local binary feature learning for face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence.

[18] H. Guan, W. A. Smith, Brisks: Binary features for spherical images on a geodesic grid, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4516–4524.

[19] H. Liu, R. Wang, S. Shan, X. Chen, Learning multifunctional binary codes for both category and attribute oriented retrieval tasks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3901–3910.

[20] Y. Duan, J. Lu, Z. Wang, J. Feng, J. Zhou, Learning deep binary descriptor with multi-quantization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1183–1192.

[21] H. Jain, J. Zepeda, P. Pérez, R. Gribonval, Subic: A supervised, structured binary code for image search, arXiv preprint arXiv:1708.02932.

[22] Y. Shen, L. Liu, L. Shao, J. Song, Deep binaries: Encoding semantic-rich cues for efficient textual-visual cross retrieval, arXiv preprint arXiv:1708.02531.

[23] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary robust independent elementary features, in: Proceed. of Europ. Conf. on Comp. Vision, Springer, Crete, Greece, 2010.

[24] T. Ojala, M. Pietikäinen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern recognition 29 (1) (1996) 51–59.

[25] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: an efficient alternative to SIFT or SURF, in: Proceed. of IEEE Intern. Conf. on Comp. Vision, Barcelona, Spain, 2011.

[26] S. Leutenegger, M. Chli, R. Y. Siegwart, BRISK: Binary robust invariant scalable keypoints, in: Proceed. of IEEE Intern. Conf. on Comp. Vision, Barcelona, Spain, 2011.

[27] A. Alahi, R. Ortiz, P. Vandergheynst, FREAK: Fast retina keypoint, in: Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec., Providence, Rhode Island, USA, 2012.

[28] L. Baroffio, A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, Briskola: BRISK optimized for low-power ARM architectures, in: Proceed. of IEEE Intern. Conf. Image Proc., Paris, France, 2014.

[29] T. Trzcinski, M. Christoudias, P. Fua, V. Lepetit, Boosting binary keypoint descriptors, in: Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec., Portland, Oregon, USA, 2013.

[30] E. Rosten, T. Drummond, Fusing points and lines for high performance tracking, in: Proceed. of IEEE Intern. Conf. on Comp. Vision, Beijing, China, 2005.

[31] E. Mair, G. D. Hager, D. Burschka, M. Suppa, G. Hirzinger, Adaptive and generic corner detection based on the accelerated segment test, in: Proceed. of Europ. Conf. on Comp. Vision, Springer, Crete, Greece, 2010.

[32] J.-M. Morel, G. Yu, ASIFT: A new framework for fully affine invariant image comparison, SIAM Journal on Imaging Sciences 2 (2) (2009) 438–469.

[33] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, Intern. J. of Comp. Vision 60 (1) (2004) 63–86.

[34] Y. Pang, W. Li, Y. Yuan, J. Pan, Fully affine invariant SURF for image matching, Neurocomputing 85 (2012) 6–10.

[35] H. Ling, D. W. Jacobs, Deformation invariant image matching, in: Proceed. of IEEE Intern. Conf. on Comp. Vision, Beijing, China, 2005.

[36] F. Moreno-Noguer, Deformation and illumination invariant feature point descriptor, in: Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec., Colorado Springs, USA, 2011.

[37] C. Wu, B. Clipp, X. Li, J.-M. Frahm, M. Pollefeys, 3D model matching with viewpoint-invariant patches (VIP), in: Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec., Anchorage, Alaska, USA, 2008.

[38] M. Karpushin, G. Valenzise, F. Dufaux, Local visual features extraction from texture+depth content based on depth image analysis, in: Proceed. of IEEE Intern. Conf. Image Proc., Paris, France, 2014.

[39] T.-W. R. Lo, J. P. Siebert, Local feature extraction and matching on range images: 2.5D SIFT, Comp. Vision and Image Understanding 113 (12) (2009) 1235–1250.

[40] B. Steder, R. B. Rusu, K. Konolige, W. Burgard, Point feature extraction on 3D range scans taking into accountobject boundaries, in: Proceed. of IEEE Intern. Conf. on Rob. and Autom., Shanghai, China, 2011.

[41] B. Lin, F. Zhao, T. Tamaki, F. Wang, L. Xiao, SIPF: Scale invariant point feature for 3D point clouds, in: Proceed. of IEEE Intern. Conf. Image Proc., Qubec City, Canada, 2015.

[42] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: Proceed. of Europ. Conf. on Comp. Vision, Springer, Crete, Greece, 2010.

[43] R. B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3D registration, in: Proceed. of IEEE Intern. Conf. on Rob. and Autom., Kobe, Japan, 2009.

[44] A. Ramisa, G. Alenya, F. Moreno-Noguer, C. Torras, FINDDD: A fast 3D descriptor to characterize textiles for robot manipulation, in: Proceed. of IEEE Intern. Conf. on Intellig. Robots and Systems, Tokyo, Japan, 2013.

[45] R. B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3d registration, in: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 3212–3217.

[46] A. E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, IEEE Trans. Pattern Anal. Machine Intell. 21 (5) (1999) 433–449.

[47] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, T. Funkhouser, 3DMatch: Learning local geometric descriptors from RGB-D reconstructions, in: CVPR, 2017.

[48] F. Tombari, S. Salti, L. Di Stefano, A combined texture-shape descriptor for enhanced 3D feature matching, in: Proceed. of IEEE Intern. Conf. Image Proc., Brussels, Belgium, 2011.

[49] E. R. do Nascimento, G. L. Oliveira, A. W. Vieira, M. F. Campos, On the development of a robust, fast and lightweight keypoint descriptor, Neurocomputing 120 (2013) 141–155.

[50] K. Koser, R. Koch, Perspectively invariant normal features, in: Proceed. of IEEE Intern. Conf. on Comp. Vision, Rio de Janeiro, Brazil, 2007.

[51] D. Gossow, D. Weikersdorfer, M. Beetz, Distinctive texture features from perspective-invariant keypoints, in: Proceed. of IEEE Intern. Conf. on Pattern Rec., Tsukuba, Japan, 2012.

[52] M. Karpushin, G. Valenzise, F. Dufaux, Improving distinctiveness of BRISK features using depth maps, in: Proceed. of IEEE Intern. Conf. Image Proc., Québec city, Canada, 2015.

[53] A. Zaharescu, E. Boyer, K. Varanasi, R. Horaud, Surface feature detection and description with applications to mesh matching, in: Proceed. of IEEE Intern. Conf. on Comp. Vision and Pattern Rec., Miami, USA, 2009.

[54] M. Karpushin, G. Valenzise, F. Dufaux, A scale space for texture+depth images based on a discrete Laplacian operator, in: IEEE Intern. Conf. on Multimedia and Expo, Torino, Italy, 2015.

[55] M. Karpushin, G. Valenzise, F. Dufaux, Keypoint detection in RGBD images based on an efficient viewpoint-covariant multiscale representation, in: Proceed. of Europ. Sign. Proc. Conf., EURASIP, Budapest, Hungary, 2016.

[56] M. Karpushin, G. Valenzise, F. Dufaux, Keypoint detection in RGBD images based on an anisotropic scale space, IEEE Trans. Multimedia 18 (9) (2016) 1762–1771.

[57] R. B. Rusu, S. Cousins, 3D is here: Point cloud library (PCL), in: Proceed. of IEEE Intern. Conf. on Rob. and Autom., Shanghai, China, 2011.

[58] C. Harris, M. Stephens, A combined corner and edge detector., in: Alvey vision conference, Vol. 15, Manchester, UK, 1988, p. 50.

[59] M. Brown, D. G. Lowe, Invariant features from interest point groups., in: Proceed. of British Machine Vision Conf., Cardiff, UK, 2002.

[60] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, in: Proceed. of Intern. Conf. on Multimedia, MM '10, ACM, New York, USA, 2010. doi:10.1145/1873951.1874249.

[61] M. Karpushin, G. Valenzise, F. Dufaux, An image smoothing operator for fast and accurate scale space approximation, in: Proceed. of IEEE Intern. Conf. Acoust., Speech and Sign. Proc., Shanghai, China, 2016.

[62] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of RGB-D SLAM systems, in: Proc. of the Intern. Conf. on Intelligent Robot Systems, Vilamoura, Algarve, Portugal, 2012.

[63] M. Agrawal, K. Konolige, M. R. Blas, Censure: Center surround extremas for realtime feature detection and matching, in: Proceed. of Europ. Conf. on Comp. Vision, Springer, Marseille, France, 2008.

[64] P. J. Besl, N. D. McKay, Method for registration of 3-D shapes, in: Robotics-DL tentative, International Society for Optics and Photonics, 1992, pp. 586–606.