IMPROVING DISTINCTIVENESS OF BRISK FEATURES USING DEPTH MAPS

Maxim Karpushin, Giuseppe Valenzise, Frédéric Dufaux

Institut Mines-Télécom; Télécom ParisTech; CNRS LTCI

ABSTRACT

Binary local descriptors are widely used in computer vision thanks to their compactness and robustness to many image transformations such as rotations or scale changes. However, more complex transformations, like changes in camera viewpoint, are difficult to deal with using conventional features due to the lack of geometric information about the scene. In this paper, we propose a local binary descriptor which assumes that geometric information is available as a depth map. It employs a local parametrization of the scene surface, obtained through depth information, which is used to build a BRISK-like sampling pattern intrinsic to the scene surface. Although we illustrate the proposed method using the BRISK architecture, the obtained parametrization is rather general and could be embedded into other binary descriptors. Our simulations on a set of synthetically generated scenes show that the proposed descriptor is significantly more stable and distinctive than popular BRISK descriptors under a wide range of viewpoint angle changes.

Index Terms— binary descriptor; BRISK; texture+depth; matching score; distinctiveness.

1. INTRODUCTION

Increasing availability of texture+depth content allows for new approaches to classic computer vision problems. This kind of visual content consists of a 2D intensity image (*texture*) and a depth image, representing the distance of each pixel from the camera plane (*depth map*). The depth map opens large possibilities for geometrical scene reasoning, to the same extent as the binocular human vision gives a perception of geometry to the cognitive level of the human visual system. Thus, a number of problems, such as object recognition, object tracking, visual odometry, content-based image retrieval, may be treated more efficiently in case when such a geometrical information is available.

In this work, we address the image matching problem, consisting in deciding whether two given images represent a similar content, and establishing correspondences between objects represented in the images. Local visual features (local image descriptors) form the most used family of approaches to image matching. They may be split into two groups following the nature of resulting signatures: conventional descriptors, such as SIFT [1], SURF [2], GLOH [3]; and binary descriptors BRISK [4], ORB [5], BRIEF [6]. The first group offers the descriptors represented by high dimensional numeric vectors. The matching is performed by computing Euclidean distances between descriptors: when the images are similar, a large number of pairs of "close" descriptors must be found in corresponding feature sets. The second group of methods arises as a computationally efficient alternative to the first one. The efficiency raises not only on the descriptor computation stage, but on the matching stage as well, as the descriptors in this case are binary strings and may be compared via Hamming distance which is very fast to compute. At the same time, the descriptive power of such binary sequences is highly competitive to the non-binary descriptors, which is confirmed in different works ([4], [7]).

The contribution of this paper is a binary descriptor for texture+depth content, designed to be robust under rigid scene transformations (camera position changes, out-of-plane object rotations). Our descriptor uses the same sampling pattern as BRISK, but it is transformed in a specific way to meet the scene geometry, which allows for improved stability under rigid transformations. The core of our method consists in parametrizing the scene surface defined by the depth map locally at each keypoint using a *geodesic polar parametrization*. The intensity image (texture map) is then sampled accordingly to compute the descriptor.

The rest of the paper is organized as follows. The related work on local features is presented in Section 2. In Section 3, we introduce the descriptor computation process. In Section 4 we present evaluation results of the designed descriptor. Finally, Section 5 concludes the paper.

2. RELATED WORK

All the feature extractors widely used in practice, such as SIFT-like approaches or BRISK, are invariant to in-plane translations, in-plane rotations and scale changes. However, in most realistic use cases the reciprocal camera-objects movements in the scene are not restricted by these transformations. As an arbitrary object movement in 3D space may be modeled through an in-plane translation, an in-plane rotation, an out-of-plane rotation and a scale change, advanced methods are often involved into descriptor computation process to deal mainly with out-of-plane rotations.

Similarly to the invariance to in-plane rotations, one general idea consists in a local normalization before the descriptor computation, i.e., each local descriptor patch is properly warped. Classic affine-invariant detectors [8] estimate an elliptical frame per keypoint that matches a visual detail (only the intensity image is used), and the patch then undergoes an affine transformation in such a way that the ellipse is transformed into a circle. Their performance is limited for moderate out-of-plane rotations (up to 40°) [1], and the resulting descriptors may be less distinctive due to the large class of normalizing transformations [9]. A local patch normalization for texture+depth content based on planar approximation is proposed in our previous work [9]. However, that method is not suitable when a corner detector is used for keypoint detection, since non-planar keypoints are rejected before descriptor computation. Several other methods, e.g., [10] and [11], use similar normalization procedures, but they operate with range images only (generated features describe the scene geometry only, no texture data is used).

An alternative to normalization consists in *simulation* of affine transformations, i.e., generating descriptors for a set of warped versions of the input image. This idea is employed in ASIFT [12]. Demonstrating a very promising performance, this approach is lim-

ited to applications where both images are provided explicitly, as it does not produce a compact set of relevant descriptors for a given image.

In our method, we use a local geodesic polar surface parametrization. A similar representation has been previously used in [13] for the case of mesh-based face detection. Our parametrization differs from that one in the computation of the angular component.

3. PROPOSED METHOD

We propose to get an approximate description of the object surface texture leveraging the scene geometry learned from the depth map. Up to sampling effects and illumination variations, the object texture does not depend on the reciprocal camera-object orientation, so that the content description based on this becomes much more stable to arbitrary rotations and translations.

A conventional non-binary descriptor, such as SIFT or SURF, is hardly applicable for this kind of description, because it assumes a planar image patch per keypoint. If the keypoint area on the surface is not locally planar (e.g. an object corner), there is no transformation that maps isometrically the texture of such an area to a planar image. For this reason, normalization-based approaches may fail in this case. However, if the descriptor requires to sample the image only in a few points (no "continuous" image patch is assumed), which is the case of the binary descriptors, these points may be distributed over the surface like they are attached to it. This is illustrated in Fig. 1. This technique allows to make the description intrinsic to the surface texture. Then, after the image is evaluated in properly distributed sampling points, exactly the same technique may be used to compute the descriptor, i.e., a binary string is formed through pairwise comparisons of obtained samples.

Up to comparison of different binary descriptors in [7], BRISK features [4] demonstrate better overall results. For this reason we take BRISK as the base to implement the proposed idea. As any other conventional feature extraction algorithm, BRISK consists of two separated stages: keypoint detection and descriptor computation. The first stage is based on FAST detector [14] applied to the Gaussian scale space in order to achieve scale invariance. The descriptor computation consists in a set of pairwise comparisons of values obtained by sampling the image according to the pattern in Fig. 1a. As in our method we proceed similarly, we present necessary details in the following.

In this work, we use the original keypoint detection algorithm. After the keypoints are detected, we first compute the local polar parametrization at each keypoint, i.e. we look for radial and angular coordinate of each pixel, that are intrinsic to the scene surface.

3.1. Local parametrization: radial component

Let $I(u,v)$ denote the intensity image, $D(u,v)$ the depth map, H and W their heights and widths in pixels, and ω the horizontal angle of view of the camera and the depth sensor. Using the pinhole camera model we set up the following global parametrization of the scene:

$$r(u,v) = \begin{pmatrix} 2u\tan\frac{\omega}{2} \\ 2v\frac{H}{W}\tan\frac{\omega}{2} \\ 1 \end{pmatrix} D(u,v). \tag{1}$$

This global parametrization will be used to compute the desired local polar parametrization at each keypoint.

For a given keypoint centered at (u_0, v_0), we first compute the distances $\rho(u,v)$ from (u_0,v_0) to other pixels applying the fast marching algorithm [15], allowing to compute efficiently a map of geodesic distances from a given point of a surface to other points. Fast marching is a family of numerical methods solving the Eikonal equation $\|\nabla u\| = F$ in one sweep, i.e., by simulating a front propagation through the image starting from a given source point. This technique is perfectly adapted to our needs, as we do not have to process the whole image but the keypoint neighborhood only.

The fast marching is started at the keypoint center and stopped when a certain limiting distance, corresponding to the keypoint scale, is reached (this distance is further referred to as *geodesic keypoint scale* σ_g). The "keypoint area" may thereby be defined as $M = \{(u,v) : \rho(u,v) < \sigma_g\}$. The resulting geodesic distances to the keypoint center are intrinsic to the scene and do not depend on the viewpoint position. Thus, the image resulting from the fast marching application gives us directly the radial component of the parametrization we are looking for.

Geodesic keypoint scale σ_g, that limits the fast marching process, may be seen as the characteristic keypoint area size expressed in scene spatial units. It is related to the sphere radius that surrounds the keypoint area, expressed in these units. For a keypoint of scale σ, the corresponding radius is given by the following formula derived from the pinhole camera model:

$$R = \sigma D(u_0,v_0)\frac{2\tan\frac{\omega}{2}}{W} \tag{2}$$

In our tests we set σ_g equal to $6R$. This determines the scaling of the sampling pattern in function of the keypoint scale. This value is set experimentally and is reasonable in comparison to the patch extents of other descriptors; larger extent will require more time to compute the descriptor, where smaller values cause distinctiveness losses.

3.2. Local parametrization: angular component

The estimation of the angular component is more difficult. Differently to the polar geodesic parametrization in [13], we limit ourselves to an approximation, that is reasonable due to the locality and using the depth map but not an arbitrary mesh.

In a nutshell, we approximate the angular coordinate of a given point in M using precomputed values from a set of points forming a closed curve around the keypoint center. So, we first extract a level curve on the geodesic distance map $\rho(u,v)$, i.e. an oriented closed contour $C = \{(u,v) : \rho(u,v) \approx a\sigma_g\} = \{C_i\}_{i=1}^n$, where $a < 1$ is a constant. At the same time, we compute the spatial length of C by summing up the spatial distances between neighboring points. During this summation, we keep the array of cumulated lengths

$$L_k = \sum_{i=1}^{k} \|r(C_i) - r(C_{i+1})\|, k = 1,..,n. \tag{3}$$

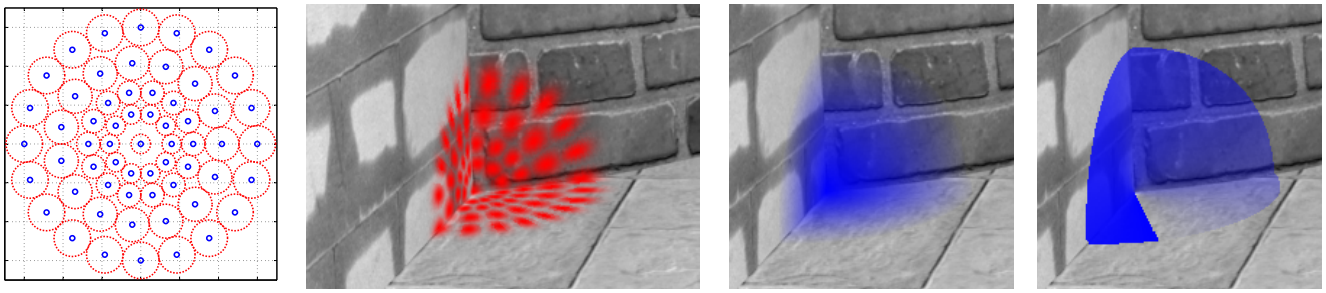
By normalizing L_k to the interval $[0, 2\pi)$ we get the "angles" ϕ_k of points of the curve C. The angular coordinate of any other point of M is then estimated by selecting that of the point in C minimizing the angle $\alpha(\vec{x},\vec{y}) = \arccos\left(\frac{(\vec{x},\vec{y})}{\|\vec{x}\|\|\vec{y}\|}\right)$ between corresponding two vectors from keypoint center:

$$i^* = \arg\min_i \alpha\left(r(u,v) - r(u_0,v_0), r(C_i) - r(u_0,v_0)\right) \tag{4}$$

$$\phi(u,v) = \phi_{i^*} \tag{5}$$

In our tests, we used $a = 0.8$, so that the reference curve C cuts the keypoint area M in two roughly equal parts in terms of number of points.

The two computed components $\rho(u,v)$ and $\phi(u,v)$, that form the local surface parametrization, are illustrated in Fig. 1c and Fig. 1d.



(a) BRISK sampling pattern (b) Exemple of distribution of the sampling points (c) Radial component $\rho(u, v)$ (d) Angular component $\phi(u, v)$

Fig. 1: Original BRISK sampling pattern for a keypoint of a unit scale [4], and an example of its distribution over the scene surface for a keypoint centered at the corner. The corresponding local parametrization is shown on images (c) and (d).

3.3. Descriptor computation

Following the BRISK architecture, we now need to smooth the image locally at each sampling point. Working in polar coordinates, we propose a "polar Gaussian kernel", a naive extension of the classic bi-dimensional Gaussian kernel to the polar coordinates, i.e. a function providing square-exponential decreasing, but in radial and angular sense. To give its analytic formulation, let us study the sampling pattern in Fig. 1a in more details.

This pattern may be splitted radially into 5 layers. The first layer consists of the center point only, each following layer contains a set of points with a constant radius and equally spaced angles. Let take a layer l having n_l points, and a point number k. Let r_l be the layer radius and s_l the associated layer scale (i.e. scale of sampling points on that layer). We define the smoothing kernel corresponding to the selected point of the layer as follows.

$$K_{l,k}(\rho, \phi) = \exp\left(-\frac{(\rho - r_l)^2}{2s_l^2} - \frac{\left(\mathrm{mod}(\phi - \frac{k-1}{n_l})\frac{n_l r_l}{4\pi}\right)^2}{2s_l^2} \right) \tag{6}$$

We denote by mod a function that wraps the angle in radians (i.e. modulo 2π).

The kernels for different (l, k) values are illustrated in Fig. 1b.

The response at the selected sample point is then given by

$$S_{l,k} = \frac{\sum_M K_{l,k}(\rho(u, v), \phi(u, v))I(u, v)}{\sum_M K_{l,k}(\rho(u, v), \phi(u, v))}. \tag{7}$$

$S_{l,k}$ for all l and k gives us the required sample values. To end up with a binary descriptor, we then proceed in a very similar way to BRISK. In a nutshell,

- we estimate the characteristic pattern direction using the long-distance sample pairs exactly as it is done in BRISK,
- we shift the angular component $\phi(u, v)$ in such a way that the characteristic direction becomes zero,
- we sample the image again with shifted $\phi(u, v)$, and (d) we compute the descriptor using the short-distance sample pairs (comparing the intensity values).

Long and short-distance pairs (sample indexes in the sampling pattern) do not depend on the content and are precomputed as in the original BRISK algorithm.

4. EXPERIMENTS AND DISCUSSION

We test our approach on a synthetic texture+depth dataset, containing significant viewpoint position changes (3 scenes, 70 images of 960*540 pixels, see [9] for image examples). We used a classic evaluation procedure from [3, 16], based on pairwise matching of different images of a given scene is performed. Specifically,

- we extract the features of each image from a given pair,
- for each descriptor in the first set, we look for the closest descriptor in the second set (in case of binary features this is the one that minimizes the Hamming distance),
- for each feature we compute a sphere in 3D space (its center position in global 3D coordinates and its radius using formula (2)), that "contain" the keypoint; the necessary ground-truth data is provided within the dataset,
- similarly to [16] we use *overlap error* to determine whether the two features of a given pair describe the same area of 3D scene or not, i.e. a pair of keypoints i and j is labeled as a *true match* if corresponding spheres S_i and S_j satisfy

$$|S_i \cap S_j| \geq (1 - \epsilon)|S_i \cup S_j|. \tag{8}$$

Otherwise a *false match* is got.

We set the error threshold ϵ equal to 0.5. In our tests we compute the volumetric overlap, but not planar, as proposed in [3, 16] (i.e. intersection of spheres instead ellipses), because the scenes we use are not entirely planar.

The proposed method is compared to the original BRISK (authors implementation is used). For completeness we also add to the comparison SIFT descriptor (implemented in VLFeat library [17]). However, for the consistency of experiments, all the descriptors are tested with the original BRISK detector, even if in practice SIFT typically uses its own detector.

We first compute the *matching score*, i.e., a portion of correctly matched features to the maximum possible number of matches, as proposed in [16]. In this case, a match is simply a pair containing a descriptor from the first image and its nearest descriptor from the second image. But in practice, such a set may contain a lot of *false matches*. To reduce their amount, a certain score is typically assigned to each pair, and then a threshold is applied. The ability of descriptor to preserve *true matches* (so, to keep the *matching score*) and to reject *false matches* when increasing the threshold, represents its distinctiveness. The distinctiveness is typically evaluated through receiver operating characteristics (ROC) that shows the portions of true

and false matches kept for different thresholds. We computed ROC curves similarly to [7]: a statistically significant number of closest descriptors pairs was selected, cumulated histograms for *true* and *false* matches in function of *pair scores* were computed to plot ROC curves. The curves are parametrized by the threshold applied to pair scores. The *pair score* for binary descriptors is simply the Hamming distance. For the SIFT descriptor we used the originally proposed distances ratio to the first closest and second closest descriptor (see [1] for details), as it gives better results than a simple inter-descriptor distance. Finally we compute areas under ROC curves (AUC), splitting the matches in three groups for limited (up to 30°), moderate (30°–60°) and large (more than 60°) viewpoint angle changes to evaluate the descriptors stability for different ranges of out-of-plane rotations.

The geometrical correction performed in our descriptor allows to match the image patches viewed under a large spectrum of angles of view, that nor BRISK neither SIFT descriptor could achieve. This is confirmed by higher matching scores, as shown in Fig. 3, especially for high-detailed texture (*Graffiti*). Moreover, our descriptor outperforms the other methods in terms of ROC and AUC as presented in Fig. 2 and Fig. 4. The limited performance of SIFT descriptor is mainly explained by using the inadapted corner BRISK detector instead of the original blob detector.

5. CONCLUSION

In this paper we designed a local descriptor for texture+depth visual content aimed at better stability under rigid scene transformations, such as out-of-plane rotations or significant viewpoint position changes. The descriptor is based on BRISK sampling pattern, which is projected on the scene surface using the depth map, so that the produced binary signature describes the object texture from the observed intensity image intrinsically to the scene geometry. The proposed descriptor is evaluated on an artificial dataset, and demonstrated a significant improvement in terms of ROC, AUC and matching score compared to the standard BRISK features.

The result may be further improved by using a detector that computes salient visual points taking into account the scene geometry. The design of such a detector makes part of our future work. Moreover, with this local content description approach, some more complex transformations may be addressed, such as isometric non-rigid surface deformations. In the future, we will address the problem of designing a complete local feature extraction pipeline from texture+depth images with the ambitious goal of stability under such complex scene deformations.

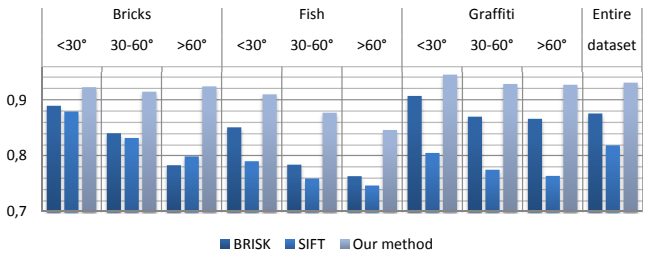


Fig. 2: Areas under ROC curves obtained on test sequences for different ranges of out-of-plane rotations, and on the entire dataset (corresponding curves are presented in Fig. 4).

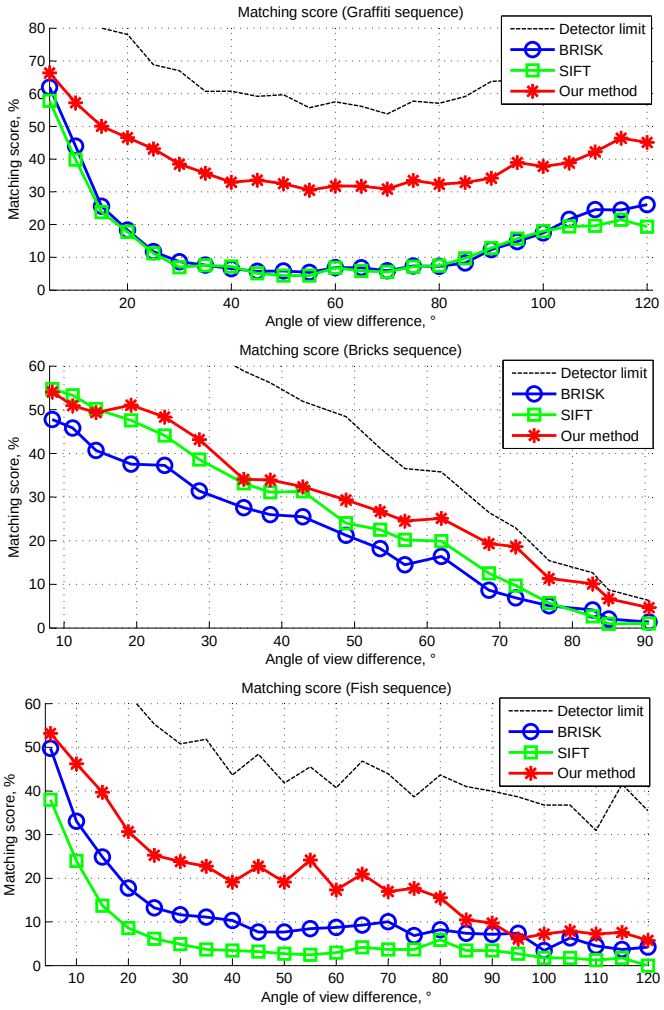


Fig. 3: Matching scores obtained on different sequences. Black curves represents detector limitations, i.e. numbers of repeated keypoints (*repeatability*).

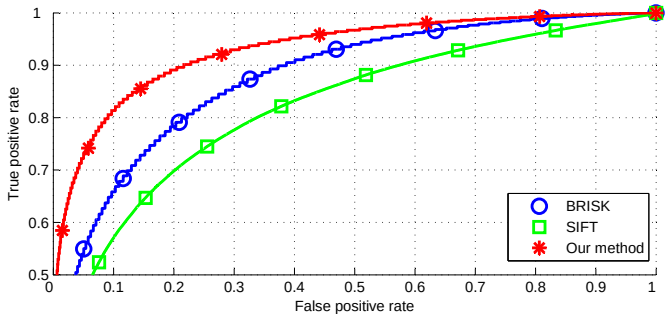


Fig. 4: Receiver operating characteristics obtained on the entire dataset.

6. REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[4] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision (ICCV), 2011*. IEEE, 2011, pp. 2548–2555.

[5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision, 2011*. IEEE, 2011, pp. 2564–2571.

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 778–792.

[7] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla, "Evaluation of low-complexity visual feature detectors and descriptors," in *2013 18th International Conference on Digital Signal Processing (DSP)*. IEEE, 2013, pp. 1–7.

[8] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.

[9] M. Karpushin, G. Valenzise, and F. Dufaux, "Local visual features extraction from texture+depth content based on depth image analysis," in *2014 20th International Conference on Image Processing (ICIP)*. IEEE, 2014.

[10] T.-W. R. Lo and J. P. Siebert, "Local feature extraction and matching on range images: 2.5D SIFT," *Computer Vision and Image Understanding*, vol. 113, no. 12, pp. 1235–1250, 2009.

[11] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "Point feature extraction on 3d range scans taking into account object boundaries," in *2011 IEEE International Conference on Robotics and automation (ICRA)*. IEEE, 2011, pp. 2601–2608.

[12] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[13] I. Mpiperis, S. Malassiotis, and M. G. Strintzis, "3-D face recognition with the geodesic polar representation," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 537–547, 2007.

[14] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 183–196.

[15] A. Spira and R. Kimmel, "An efficient solution to the eikonal equation on parametric manifolds," *Interfaces and Free Boundaries*, vol. 6, no. 3, pp. 315–328, 2004.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International journal of computer vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[17] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms (2008)," 2012.