HAL
open science

Depth-Assisted Rectification for Real-Time Object Detection and Pose Estimation

J Lima, F Simões, Hideaki Uchiyama, Veronica Teichrieb, Eric Marchand

To cite this version:

J Lima, F Simões, Hideaki Uchiyama, Veronica Teichrieb, Eric Marchand. Depth-Assisted Rectification for Real-Time Object Detection and Pose Estimation. Machine Vision and Applications, 2016, 27 (2), pp.193-219. 10.1007/s00138-015-0740-8 . hal-01233046

HAL Id: hal-01233046

https://inria.hal.science/hal-01233046

Submitted on 24 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

Depth-Assisted Rectification for Real-Time Object Detection and Pose Estimation

João Paulo Silva do Monte Lima · Francisco Paulo Magalhães Simões · Hideaki Uchiyama · Veronica Teichrieb · Eric Marchand

Received: date / Accepted: date

Abstract RGB-D sensors have become in recent years a product of easy access to general users. They provide both a color image and a depth image of the scene and, besides being used for object modeling, they can also offer important cues for object detection and tracking in real-time. In this context, the work presented in this paper investigates the use of consumer RGB-D sensors for object detection and pose estimation from natural features. Two methods based on depth-assisted rectification are proposed, which transform features extracted from the color image to a canonical view using depth data in order to obtain a representation invariant to rotation, scale and perspective distortions. While one method is suitable for textured objects, either planar or non-planar, the other method focuses on texture-less planar objects. Qualitative and quantitative evaluations of the proposed methods are performed, showing that they can obtain better results than some existing methods for object detection and pose estimation, especially when dealing with oblique poses.

Keywords Object Detection · Natural Features Tracking · Computer Vision · RGB-D Sensor

J. Lima
Departamento de Estatística e Informática (DEINFO)
Universidade Federal Rural de Pernambuco (UFRPE)
Recife/PE, Brazil
E-mail: jpsml@deinfo.ufrpe.br

J. Lima, F. Simões, V. Teichrieb
Voxar Labs
Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife/PE, Brazil

H. Uchiyama, E. Marchand
INRIA Rennes Bretagne-Atlantique
Rennes, France

1 Introduction

Computer vision systems commonly sense the real world by detecting planar fiducial markers placed around it. However, in many applications the use of such kind of markers is undesirable. In these cases, a better way to sense the world would be to detect and track real objects using natural features of the scene.

In recent years, computer vision applications have benefited from the advent of low cost RGB-D consumer devices [10]. These devices are commonly used in human body detection and tracking for user interaction purposes. RGB-D sensors are able to provide in real-time, besides a color image (RGB channels) of the scene, another image in which each pixel value corresponds to the distance from the scene objects to the camera. Such image is named depth image (D channel). There are different types of RGB-D sensors, such as stereo cameras [56] and projected texture stereo [26]. Nevertheless, this paper focuses on existing consumer RGB-D sensors. The first consumer RGB-D devices available for mass market provided the RGB image using a standard color camera and computed the depth image using infrared (IR) camera and projector. The IR projector is used to project known patterns that are recognized by the IR camera. The depth is then estimated by triangulation between camera and projector. Newer consumer RGB-D cameras combine a standard RGB sensor with a time-of-flight (ToF) sensor that provides a depth image of the scene. The ToF camera computes depth information by measuring the time that it takes to a light pulse to travel from the camera to an object and back. The use of RGB-D consumer devices for object detection and pose estimation has grown significantly over the last years [21, 30]. The color and depth images from RGB-D cameras can be employed to obtain 3D models

of the objects to be detected and also provide useful information at runtime for accomplishing better results when compared to techniques that use only RGB data.

In this context, this paper presents two novel real-time object detection and pose estimation methods that use natural features and consumer RGB-D sensors. The developed techniques are based on depth-assisted rectification, which consists in obtaining a representation invariant to rotation, scale and perspective distortions by transforming natural features extracted from the color image to a canonical view using depth data. The first method is named Depth-Assisted Rectification of Patches (DARP) and is suitable for textured objects, either planar or non-planar. The second method is named Depth-Assisted Rectification of Contours (DARC) and focuses on texture-less planar objects. The methods are straightforwardly designed, providing good results without having to use more complex approaches. Such simplicity is obtained by using an RGB-D camera, which is a more complex but very popular sensor.

The contributions of this paper are: (1) a patch rectification method that uses depth information to obtain a perspective and scale invariant representation of keypoints; (2) a framework for rectifying, matching and estimating the pose of contours extracted from an RGB image using depth data, being invariant to rotation, scale and perspective deformations; (3) an approach for determining which depth-assisted rectification method is more suitable for detecting a given object; (4) a strategy for using both patch and contour depth-assisted rectification techniques together; (5) a frame-to-frame tracking method using the developed depth-assisted rectification techniques; (6) qualitative and quantitative evaluations regarding pose estimation quality of the developed methods in comparison with existing techniques; (7) runtime analyses of the developed techniques in comparison with existing detection methods, verifying their compliance to real-time constraints.

This paper is organized as follows. Sect. 2 presents works related to DARP and DARC and how these methods differ from them. Sect. 3 presents the DARP method, which makes use of depth information for rectifying patches around interest points in the color image. Sect. 4 presents the DARC method, which rectifies contours extracted from the color image using depth data. Sect. 5 introduces an approach for selecting between DARP or DARC based on an image of the object to be detected. Sect. 6 presents a strategy for detecting objects using both DARP and DARC together. Sect. 7 shows how DARP and DARC can also be used for frame-to-frame tracking. Sect. 8 brings a discussion about the results obtained with DARP and DARC, comparing them with other existing object detection and pose estimation methods. Sect. 9 presents final considerations and future work.

2 Related work

In the next subsections, some existing object detection methods related to DARP and DARC are described. Subsect. 2.1 describes detection methods suitable for textured objects, which are targeted by DARP. Subsect. 2.2 details detection techniques for texture-less objects, which are handled by DARC.

2.1 Textured object detection

A common approach for detecting textured objects on images captured under different viewpoints consists in extracting local discriminative repeatable features from the images. Some of these features are only invariant to rotation, such as Harris corners [18] and FAST keypoints [49], and scale invariance is often obtained by detecting features from different levels of an image pyramid. There are some features that are invariant to both rotation and scale, like local extrema of Difference of Gaussians (DoG) [37]. Some features are also invariant to affine transformations, such as affine regions [41]. Object detection is then performed by matching features extracted from the query image to previously obtained features from template images with known pose, even if the images were obtained from significantly different viewpoints. One alternative for performing this matching is by using local descriptors, which are high dimensional vectors that describe the neighborhood around the local feature. Examples of local descriptors are SIFT [37], SURF [3], HIP [54], BRIEF [8] and rBRIEF [50]. Descriptor matching is done by nearest neighbor search based on the distance between the high dimensional vectors. Another way of matching local features is by using classifiers such as Randomized Trees [31] and Ferns [45]. They are trained beforehand using object local features with different poses.

Nevertheless, many existing keypoint matchers fail on scenarios where objects have a very oblique pose with respect to the viewing direction and suffer from severe perspective distortions. Since perspective deformations can be approximated by affine transformations for small areas, affine invariant local features can be used to generate normalized patches [41]. On the other hand, DARP can use local features that are, a priori, not affine and scale invariant, performing a posteriori projective rectification of the patches.

The ASIFT method [42] obtains a higher number of matches from perspectively distorted images by generating several affine transformed versions of both images and then finding correspondences between them using SIFT [37]. Alternatively, the DARP method is able to use solely the query and template images in order to match them. ASIFT also makes use of low-resolution versions of the affine transformed images in order to accelerate keypoint matching. Only the affine transformations that provide more matches are used to compare the images in their original resolution. The DARP technique is able to work directly with high resolution images, without needing to decrease their quality to achieve real-time keypoint matching.

In [27], Maximally Stable Extremal Region (MSER) features [40] are projectively rectified using Principal Component Analysis (PCA) and graphics hardware. However, it does not focus on real-time execution and it is designed to work with region detectors, while the DARP method works with keypoint detectors and computes rectified patches in real-time.

Patch perspective rectification is also performed in [11, 20, 22, 46]. These methods differ from DARP because they first estimate patch identity and coarse pose, and then refine the pose of the identified patch. In DARP, the patches are first rectified in order to allow estimating their identity. In addition, these methods need to previously generate warped versions of the patch for being able to compute its rectification, while DARP can rectify a patch without such constraint.

The methods described in [13, 28, 57, 58] first projectively rectify the whole image and then detect invariant features on the normalized result, while the DARP method does the opposite. In addition, [57] is designed for offline 3D reconstruction, [13, 28, 58] target only planar scenes and [13, 28] require an inertial sensor.

A method for keypoint matching of developable surfaces (such as cones or cylinders) under different viewpoints using a consumer RGB-D sensor is presented in [59]. The surfaces are first unrolled exploiting depth information and then the rectified textures are employed for keypoint detection and matching. Dealing with the rectified textures instead of the original images allows obtaining a higher number of correct matches. A similar approach is performed in [60], but without requiring the presence of particular geometric shapes by using salient directions, which are peaks in the distribution of surface normals. However, it needs to generate several salient direction rectified images in order to perform registration, thus not focusing on real-time execution.

Concurrent with this research, the techniques detailed in [38, 15] also used an RGB-D sensor to perform patch rectification using PCA. In [38], a descriptor for the patch is obtained using 2D Fourier-Mellin Transform. Nevertheless, the rectification algorithm applied is not clearly described and it is not evaluated under a real-time keypoint matching scenario. The Depth-Adaptive Feature Transform (DAFT) method is presented in [15], where the DoG detector is adapted to use depth information for obtaining scale invariant keypoints and SURF is used to describe the rectified patches. The results obtained using DARP and DAFT are compared in Sect. 8.

2.2 Texture-less object detection

Local feature descriptors such as the ones listed in the previous subsection showed to be not suitable for dealing with texture-less objects, since it is hard to obtain repeatable and discriminative features from such kind of object. Therefore, recent researches have been focused on methods that are able to detect and estimate the pose of texture-less objects.

One option for detecting texture-less objects is to perform a search over the pose space using template matching, such as in [24]. However, when the pose range increases, the processing time required by this kind of technique makes them unsuitable for real-time applications.

Most existing techniques suitable for texture-less objects need to capture several views of the target object or to generate perspective warps from reference images. The method described in [25] trains a classifier with normalized distance transform templates computed from warped versions of a reference image. It aims to detect and estimate the pose of planar targets. In [20, 22] perspective rectification is learned from warped patches in order to allow matching of local features. Dominant orientation templates are generated in [23] from a number of different viewpoints for estimating the pose of texture-less 3D objects. The approach detailed in [21] acquires RGB-D images from many views of a texture-less 3D object and makes use of 2D image gradients and 3D surfaces normals for estimating its pose. In [47], dominant orientation templates of grayscale images obtained from different viewpoints are used to estimate a coarse pose of texture-less 3D objects. The pose is then refined using RGB-D data. This method was later extended in [30] to also compute dominant orientation templates from the depth image. In addition, it demonstrates the capability of discerning objects with the same shape and texture but different sizes by exploiting depth information, which is also done by DARC. A technique described in [1] performs pose estimation based on junctions by comparing the query image with previously acquired keyframes of the

target texture-less 3D object from many views. In [12], distance transforms computed from warped versions of MSERs are used to train a classifier. This allows estimating the pose of planar contours by exploiting projective invariants, as long as the contour has at least one concavity. In contrast, the DARC technique needs only an RGB-D image of the planar object taken from a single view for estimating its pose. It also stores two or four versions of each template relative to its different orientations, without needing to generate several warps. The DARC method is comparable to the approach described in [16], which stores a single signature for each template contour. However, it makes use of projective invariants with low discriminative power, leading to potential wrong matches with background features. The technique detailed in [39] is able to detect contours by keypoint matching with a single reference image, but the keypoint descriptor used is not invariant to severe perspective distortions.

There are some other techniques in the literature that perform feature rectification for 3D registration. Methods that use a 3D reconstruction of the scene often rely on texture based local descriptors and are not adequate for texture-less objects [27, 38, 57, 58]. There are also some approaches that require the presence of inertial sensors [13, 28]. The DARC method does not need any additional sensor besides an RGB-D camera and is based on normalization of contour features, allowing pose estimation of texture-less planar targets. To the best of the authors' knowledge, there are no other methods in the literature based on RGB-D images that focus on texture-less planar object detection and 6DOF pose estimation.

3 Depth-Assisted Rectification of Patches

This section presents the DARP method, which exploits depth information available in RGB-D consumer devices to improve keypoint matching of perspectively distorted images [34, 33]. This is achieved by generating a projective rectification of a patch around the keypoint, which is normalized with respect to perspective distortions and scale. An overview of the DARP technique is illustrated in Fig. 1. In DARP, keypoints are extracted and their normal vectors on the scene surface are estimated using the depth image. Then, using depth and normal information, patches around the keypoints are rectified to a canonical view in order to remove perspective and scale distortions. The rectified patch orientation is calculated in order to obtain rotation invariance. Finally, a descriptor for the rectified patch is calculated using the assigned orientation. DARP can be used with any local feature detector and descriptor and is suitable for planar and non-planar textured scenes. In the next subsections, all steps of the DARP method are detailed.

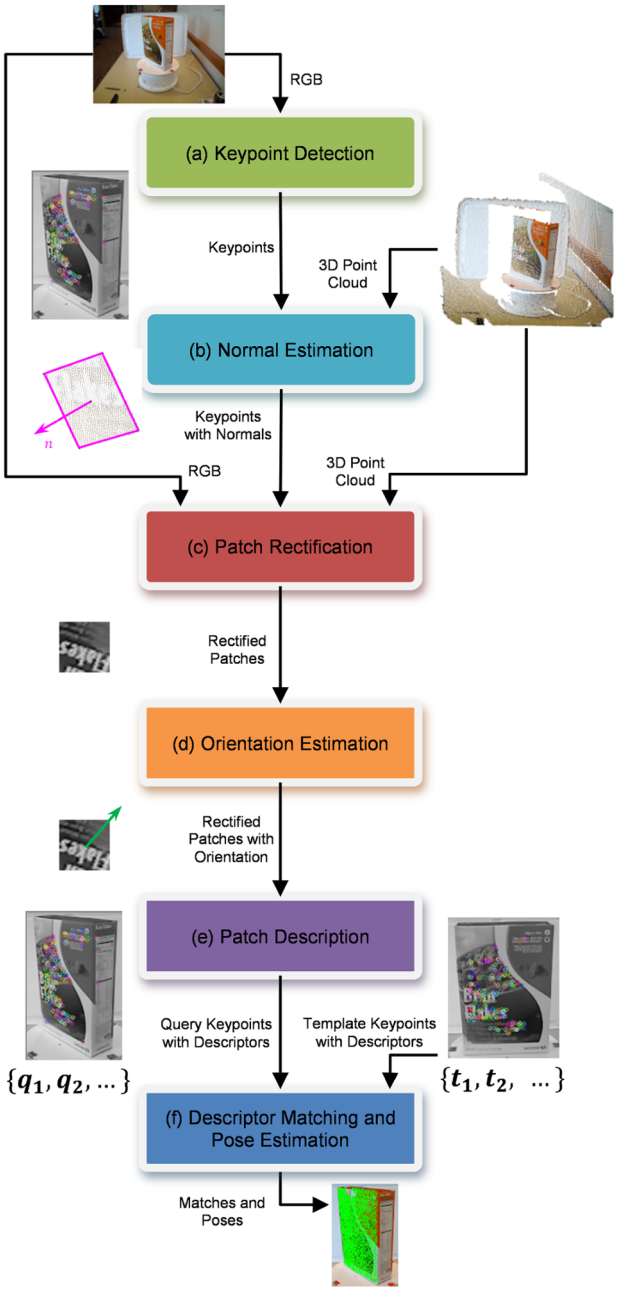


Fig. 1 DARP method overview. (a) Keypoints are detected using the RGB image. (b) Normal is computed for each keypoint using the 3D point cloud calculated from the depth image. (c) Patches are rectified using normal, RGB image and the 3D point cloud. (d) Orientation is calculated for each rectified patch. (e) A descriptor is computed for each oriented rectified patch. (f) Query keypoints descriptors are matched to template keypoints descriptors and a pose is calculated using the correspondences

Fig. 2 Keypoint detection example, where each detected keypoint is represented by a colored circle

3.1 Keypoint detection

Any keypoint detector can be used by DARP, such as Harris corners [18], FAST-9 [49] or DoG [37]. Since the patch around the keypoint is normalized a posteriori with respect to perspective distortions and scale, the detector does not have to be affine or scale invariant and the use of a scale pyramid for the input image is not mandatory. Fig. 2 illustrates keypoints detected on an input image.

3.2 Normal estimation

From the query depth image, a 3D point cloud in camera coordinates can be computed for the scene. Considering a 3D point $\mathbf{M_{cam}} = [M_x, M_y, M_z]^T$ in camera coordinates, its 2D projection $\mathbf{m} = [m_x, m_y, 1]^T$ is given by:

$$\mathbf{m} = \begin{bmatrix} f_x M_x/M_z + c_x \\ f_y M_y/M_z + c_y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} M_x/M_z \\ M_y/M_z \\ 1 \end{bmatrix}, \quad (1)$$

where f_x and f_y are the focal length in terms of pixel dimensions in the x and y direction respectively, c_x and c_y are the coordinates of the principal point and $\mathbf{K}$ is known as the intrinsic parameters matrix. Thus, rearranging the terms and considering $M_z = d$, where d is the depth of $\mathbf{m}$, the coordinates of $\mathbf{M_{cam}}$ can be obtained by:

$$\mathbf{m} = \begin{bmatrix} (m_x - c_x)d/f_x \\ (m_y - c_y)d/f_y \\ d \end{bmatrix}. \quad (2)$$

Using this point cloud, a normal vector can be estimated for a 3D point $\mathbf{M_{cam}}$ that corresponds to an extracted 2D keypoint via PCA. The centroid $\bar{\mathbf{M}}$ of all neighbour 3D points $\mathbf{M_i}$ within a radius of 3 cm of $\mathbf{M_{cam}}$ is computed. A covariance matrix is computed using $\mathbf{M_i}$ and $\bar{\mathbf{M}}$, and its eigenvectors $\{\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}$ and corresponding eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ are computed and ordered in ascending order. The normal vector to the scene surface at $\mathbf{M_{cam}}$ is given by $\mathbf{v_1}$ [5], which is depicted in Fig. 3. If needed, $\mathbf{v_1}$ is flipped to aim towards the viewing direction. Only the keypoints that have a valid normal are kept.

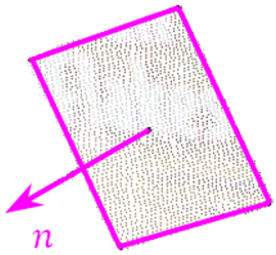


Fig. 3 Normal vector of a patch on the scene surface

3.3 Patch rectification

The next step consists in using the available 3D information to rectify a patch around each keypoint to remove perspective deformations. In addition, a scale normalized representation of the patch is obtained. This is done by computing a homography that transfers the patch to a canonical view, as illustrated in Fig. 4. Given $\mathbf{n} = [n_x, n_y, n_z]^T$ as the unit normal vector in camera coordinates at $\mathbf{M_{cam}}$, which is the corresponding 3D point of a keypoint, two unit vectors $\mathbf{n_1}$ and $\mathbf{n_2}$ that define a plane with normal $\mathbf{n}$ can be obtained by:

$$\mathbf{n_1} = \frac{1}{\|(n_z, 0, -n_x)^T\|}(n_z, 0, -n_x)^T, \quad (3)$$

$$\mathbf{n_2} = \mathbf{n} \times \mathbf{n_1}. \quad (4)$$

This is valid because it is assumed that $\mathbf{n_x}$ and $\mathbf{n_z}$ are not equal to zero at the same time, since in this case the normal would be perpendicular to the viewing direction and the patch would be not visible.

From $\mathbf{M_{cam}}$, $\mathbf{n_1}$ and $\mathbf{n_2}$, it is possible to find the corners $\mathbf{M_1}, \ldots, \mathbf{M_4}$ of the patch in the camera coordinate system. The patch size in camera coordinates should be fixed in order to allow scale invariance. The corners $\mathbf{m_1}, \ldots, \mathbf{m_4}$ of the patch to be rectified in image coordinates are the projection of the 3D points $\mathbf{M_1}, \ldots, \mathbf{M_4}$. Then, $\mathbf{m_i} = \mathbf{K}\mathbf{M_i}$, where $\mathbf{K}$ is the intrinsic parameters matrix. If the patch size in image coordinates is too small, the rectified patch will suffer degradation in image resolution, harming its description. This size is influenced by the location of the 3D point $\mathbf{M_{cam}}$ (e.g., if $\mathbf{M_{cam}}$ is too far from the camera, the patch

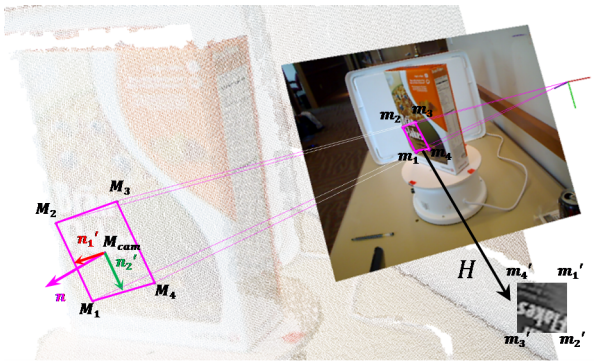


Fig. 4 Patch rectification overview. $\mathbf{M_1}$, ..., $\mathbf{M_4}$ are computed from $\mathbf{M_{cam}}$, $\mathbf{n_1}$ and $\mathbf{n_2}$. An homography $\mathbf{H}$ is computed from the projections $\mathbf{m_1}$, ..., $\mathbf{m_4}$ and the canonical corners $\mathbf{m_1}'$, ..., $\mathbf{m_4}'$[33]

size will be small). It is also directly proportional to the patch size in camera coordinates, which is determined by a constant factor k applied to $\mathbf{n_1}$ and $\mathbf{n_2}$ as follows: $\mathbf{n_1}' = k\mathbf{n_1}$ and $\mathbf{n_2}' = k\mathbf{n_2}$. The factor k should be large enough to allow good scale invariance while being small enough to give distinctiveness to the patch. In the performed experiments, different values of k were used, while the size s of the rectified patch was always set to 31.

The corners $\mathbf{M_1}, \ldots, \mathbf{M_4}$ of the patch are given by:

$$\mathbf{M_1} = \mathbf{M_{cam}} + n_1' + n_2', \tag{5}$$

$$\mathbf{M_2} = \mathbf{M_{cam}} + n_1' - n_2', \tag{6}$$

$$\mathbf{M_3} = \mathbf{M_{cam}} - n_1' - n_2', \tag{7}$$

$$\mathbf{M_4} = \mathbf{M_{cam}} - n_1' + n_2'. \tag{8}$$

The corresponding corners $\mathbf{m_1}', \ldots, \mathbf{m_4}'$ of the patch in the canonical view are:

$$\mathbf{m_1}' = (s-1, 0)^T, \tag{9}$$

$$\mathbf{m_2}' = (s-1, s-1)^T, \tag{10}$$

$$\mathbf{m_3}' = (0, s-1)^T, \tag{11}$$

$$\mathbf{m_4}' = (0, 0)^T. \tag{12}$$

From $\mathbf{m_1}, \ldots, \mathbf{m_4}$ and $\mathbf{m_1}', \ldots, \mathbf{m_4}'$, it can be computed a homography $\mathbf{H}$ that takes points of the input image to points of the rectified patch.

3.4 Orientation assignment

In order to achieve rotational invariance, the orientation of the rectified patch should be estimated. There are some different methods to obtain the dominant orientation of a patch, such as gradient orientation histogram [37], which finds dominant orientations of a patch as peaks in a histogram of quantized orientations of patch gradients, and intensity centroid [50], which computes the orientation of the patch from geometric moments. The choice of the method to compute patch orientation is often coupled to the method chosen for patch description, as both methods commonly use the same data for accomplishing their goals (such as gradients in [37] and integral images in [50]).

3.5 Patch description

The same way DARP can use any keypoint detector, it is also possible to have any patch descriptor such as SIFT [37], SURF [3], BRIEF [8] or rBRIEF [50]. In order to build a descriptor for the rectified patch, the neighborhood around the center of the patch is sampled at specific coordinates, depending on the chosen method. These coordinates are rotated with respect to the orientation computed for the rectified patch in the previous step. This way, it is possible to obtain a descriptor for each keypoint that is invariant to rotation (due to orientation normalization) and also to scale and perspective distortions (due to patch rectification).

3.6 Keypoint matching and pose estimation

For descriptor matching, a nearest neighbor search is performed to find the corresponding template descriptor for each query descriptor. Regarding pose estimation, the DLT method was used to compute object pose in the experiments performed. Homography estimation was used for planar objects, while an extrinsic parameters matrix was computed for non-planar objects. Minimization of reprojection error was used for pose refinement and the RANSAC algorithm was also applied for outliers removal.

4 Depth-Assisted Rectification of Contours

This section presents the DARC method for detection and pose estimation of texture-less planar objects using RGB-D cameras [34, 35]. It consists in matching contours extracted from the current image to previously acquired template contours. In order to achieve invariance to rotation, scale and perspective distortions, a

rectified representation of the contours is obtained using the available depth information. DARC requires only a single RGB-D image of the planar objects in order to estimate their pose, opposed to some existing approaches that need to capture a number of views of the target object. It also does not generate warped versions of the templates, which is commonly required by existing object detection techniques. Fig. 5 describes the DARC algorithm flow. First, contours are extracted from the query RGB image. Then, for each extracted contour, the 3D points that correspond to the 2D points of the contour and its inner contours are selected. The 3D contour points are used to estimate the normal and the orientation of the contour in camera coordinates. Using this information, it is possible to rectify the 3D contour to a canonical view. This rectified representation is used to perform matching between query contours and previously obtained template contours. The poses of the query contours that have a valid match are then calculated. Object detection can then be performed by detecting and estimating the pose of its contours for each frame. Each step of the DARC method is detailed in the next subsections.

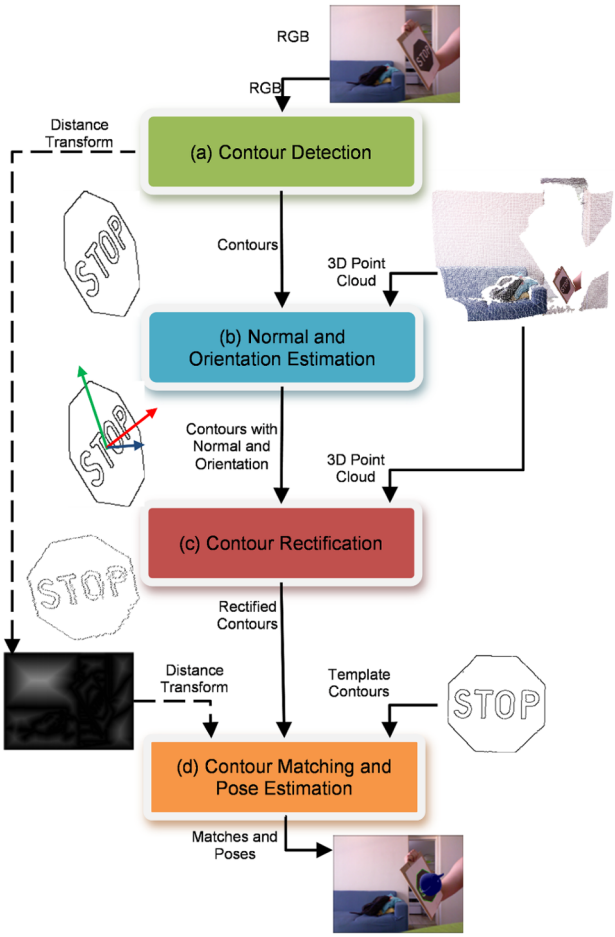


Fig. 5 DARC method overview. (a) Contours are detected using the RGB image and the distance transform is optionally computed. (b) Normal and orientation are calculated for each contour using the 3D point cloud computed from depth data. (c) Contours are rectified using normal, orientation and the 3D point cloud. (d) Rectified query contours are matched to template contours optionally using the distance transform and the poses of the query contours are obtained

4.1 Contour detection

Any contour detection method can be used by DARC and the extracted contours do not have to be affine invariant. In this work, two different approaches for detecting contours were considered: the first one is based on the Canny edge detector [9] and the second one is based on the MSER detector [40]. Each method is described next.

4.1.1 Canny contour detector

In order to obtain a binary image where contours can be extracted, the query RGB image is converted to grayscale and then the Canny edge detector is applied [9], as illustrated in Fig. 6. A dilation operator can also be applied to the binary image in order to connect broken edge segments. The algorithm described in [53] is used to extract closed contours from the binary image. Contours that have an area smaller than a threshold are discarded.

Similarly to [25], the hierarchy of contours is also exploited in order to increase their discriminative power. When dealing with a closed contour in all the following steps of the method, its inner contours are also considered as part of the parent contour representation. In the remainder of this paper, the set of points that belong to a contour or its inner contours is named *contour group*. Since more information is taken into account when contour hierarchy is used, it allows obtaining a more accurate estimation of contour rotation and also improves the measurement of similarity between two different contours. Contour hierarchy is also needed at runtime to correctly group the query contours that correspond to a previously acquired template contour group.

In addition, the distance transform is computed from the binary image with the sequential algorithm described in [6] for later use, obtaining a result similar to the one depicted in Fig. 7.

4.1.2 MSER contour detector

The Canny contour detector is very fast, but it is not robust to illumination changes, noise and blur caused by

Fig. 6 Canny contour detection example

Fig. 7 Distance transform computed from the binary image shown in Fig. 6

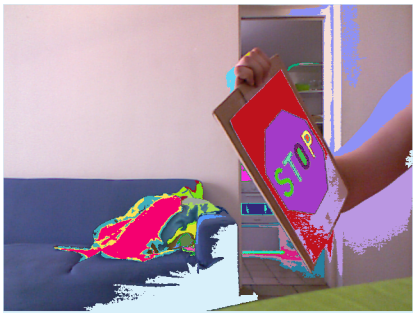


Fig. 8 MSER contour detection example, where each detected contour is filled with a solid color

very fast movements. A slower but more robust way to detect contours is to use the MSER detector [40], which is illustrated in Fig. 8. MSER uses the grayscale image obtained from the query RGB image to find stable regions with respect to thresholding over a large range of threshold values. These regions are scale and affine invariant and their boundaries can be used as contours. Since MSER deals with regions, it inherently considers the inner contours as part of an outer contour, so there is no need to use hierarchical structures to obtain contour groups as in the Canny contour detector. Actually, instead of considering only the boundary points, all the points that belong to a region detected by MSER are considered in the computation of contour normal and orientation, which is explained in the following subsection.

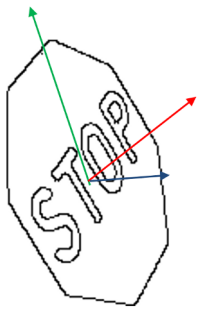


Fig. 9 Local coordinate system computed from 3D contour points using PCA

4.2 Normal and orientation estimation

From the query depth image, a 3D point cloud in camera coordinates can be computed for the scene, as discussed in Subsect. 3.2. Then, for each contour group, the corresponding 3D points $\mathbf{M_i}$ of the 2D contour points $\mathbf{m_i}$ are used to estimate the normal and orientation of the contour group via PCA. The centroid $\bar{\mathbf{M}}$ of the 3D contour points is calculated, which is invariant to affine transformations [19]. A covariance matrix is computed using $\mathbf{M_i}$ and $\bar{\mathbf{M}}$, and its eigenvectors $\{\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}\}$ and corresponding eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ are computed and ordered in ascending order. The normal vector to the contour group plane is $\mathbf{v_1}$ [5], as shown in Fig. 9. If needed, $\mathbf{v_1}$ is flipped to point towards the viewing direction. Contour group orientation is given by $\mathbf{v_2}$ and $\mathbf{v_3}$, which can be seen as the y and x axis, respectively, of a local coordinate system with origin at $\bar{\mathbf{M}}$ [5], as can be seen in Fig. 9. There are four possible orientations given by combinations of the x and y axis with different signs. It only makes sense to consider all four orientations if mirrored or transparent objects might be detected. Otherwise, only two orientations are enough, which are given by using both flipped and non-flipped $\mathbf{v_3}$ as the x axis and computing the y axis as the cross product of $\mathbf{v_1}$ and $\mathbf{v_3}$.

4.3 Contour rectification

In order to allow matching instances of the same contour group observed from different viewpoints, they are normalized to a common representation. Translation invariance is achieved by writing the coordinates of the 3D contour points $\mathbf{M_i}$ relative to the centroid $\bar{\mathbf{M}}$. Rotation invariance is obtained by aligning $\mathbf{v_3}$ and $\mathbf{v_2}$ with the x and y global axes, respectively. Since the 3D contour points $\mathbf{M_i}$ are in camera coordinates, they are scale invariant. Perspective invariance is obtained by aligning the inverse of the normal vector $\mathbf{v_1}$ to the z global axis.

Fig. 10 Rectified 3D contour points computed using Eq. 13 and 14

This way, a transformation $[\mathbf{R^r}|\mathbf{t^r}]$ can be obtained by:

$$\begin{bmatrix} \mathbf{R^r} & \mathbf{t^r} \\ \mathbf{0^T} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{v_3^T} & -\bar{\mathbf{M}} \cdot \mathbf{v_3^T} \\ \mathbf{v_2^T} & -\bar{\mathbf{M}} \cdot \mathbf{v_2^T} \\ \mathbf{v_1^T} & -\bar{\mathbf{M}} \cdot \mathbf{v_1^T} \\ \mathbf{0^T} & 1 \end{bmatrix}. \tag{13}$$

The rectified contour points $\mathbf{M_i}'$ can be computed as follows:

$$\begin{bmatrix} \mathbf{M_i}' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R^r} & \mathbf{t^r} \\ \mathbf{0^T} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{M_i} \\ 1 \end{bmatrix}. \tag{14}$$

The rectified points should lie on the xy plane ($z = 0$). Since two or four orientations given by $\mathbf{v_2}$ and $\mathbf{v_3}$ are considered, each one is used to generate a different rectification of a contour group. All these rectifications are taken into account in the matching phase. In some cases the estimated orientation is not accurate, as can be seen in the rectified contour group in Fig. 10. However, this is still sufficient for matching and pose estimation purposes.

When MSER features are used, an additional step is performed in order to rectify a binary representation of each detected region. For this, the upright bounding rectangle of the rectified contour is computed and the four corners of this rectangle are unrectified using the inverse of the $[\mathbf{R^r}|\mathbf{t^r}]$ rectifying transformation and then projected onto a binary image that represents the region. From the correspondences between the original corners and the projected corners, a homography can be computed that maps the bounding rectangle to the image, which allows obtaining a rectified version of the region, as illustrated in Fig. 11.

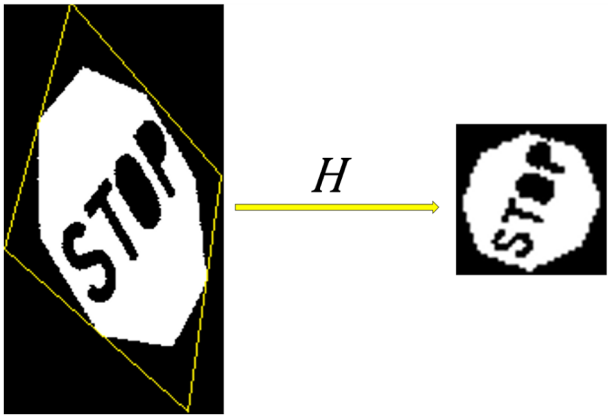


Fig. 11 Rectification of a binary representation of a detected MSER region

4.4 Contour matching and pose estimation

After being rectified, query contour groups can then be matched to a previously rectified template contour group. Two approaches were considered for contour matching and pose estimation: the first one is based on chamfer matching [2] and the second one is based on Hamming matching. The first method is used together with the Canny contour detector, while the second method is used together with the MSER contour detector.

In both approaches, some heuristics can be used to reject spurious matches. First, a match is rejected if the upright bounding rectangles of the rectified contour groups do not have a similar size. Then, it is calculated a coarse pose that maps the 3D unrectified template contour group to the 3D unrectified query contour group. Given the rotation $\mathbf{R^t}$ and translation $\mathbf{t^t}$ that rectify the template contour group and the rotation $\mathbf{R^q}$ and translation $\mathbf{t^q}$ that rectify the query contour group, the coarse pose $[\mathbf{R^c}|\mathbf{t^c}]$ is obtained by:

$$\begin{bmatrix} \mathbf{R^c} & \mathbf{t^c} \\ \mathbf{0^T} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R^q} & \mathbf{t^q} \\ \mathbf{0^T} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{R^t} & \mathbf{t^t} \\ \mathbf{0^T} & 1 \end{bmatrix}. \tag{15}$$

The 3D unrectified template contour group is transformed using the coarse pose $[\mathbf{R^c}|\mathbf{t^c}]$ and then projected onto the query image. After that, the upright bounding rectangle of the projected points is calculated and compared with the upright bounding rectangle of the 2D query contour group. If they are not close to each other or their sizes are not similar, the match is discarded.

After matching query and template contour groups using any of the methods described in the next subsections, it can be obtained several point-to-point correspondences between all the query and template contour groups that are part of the target planar object. From these correspondences, the final pose of the planar object can be computed using homography estimation together with RANSAC. One single contour group is sufficient for calculating the pose of a planar object. However, if the object is composed by several contour groups with enough discriminative power, all of them can be used for pose estimation. Using this approach, it is possible to compute the pose of the object even when some of its contours are occluded.

4.4.1 Chamfer matcher

Since rectified contour groups are invariant to rotation, scale and perspective distortions, simpler methods that do not deal with these invariants can be used to match them, such as chamfer matching [2]. The similarity between template contour group projection and 2D query contour group is given by their chamfer distance:

$$\frac{1}{\tau n}\sum_{i=0}^{n} DT^{\tau}(\mathbf{m_i^t}), \tag{16}$$

where n is the number of points in the template contour group, $\mathbf{m_i^t}$ is the i-th template contour point and DT^{τ} is the query distance transform truncated to a value τ, which was set to 20. For each query contour group, the template contour group orientation with smallest chamfer distance is marked as a candidate match.

If there is a candidate match for a given query contour group, then a refined pose of the contour group is estimated from the previously computed coarse pose $[\mathbf{R^c}|\mathbf{t^c}]$ using the Levenberg-Marquardt algorithm. The query distance transform is used to compute the reprojection error. Finally, the chamfer distance between the template contour group and query contour group is calculated using the refined pose. If it is below a threshold, then the match is considered as correct. The truncation of the distance transform to a value τ has an effect on the minimization similar to using the Tukey M-estimator.

4.4.2 Hamming matcher

The rectified binary representations obtained for MSER features can be matched by calculating their Hamming distance using a bitwise XOR operation. The percentage of black pixels on the resulting XOR image gives a measure of similarity between query and template regions.

Using a binary image representing the query region, the rectifying homography computed as in Subsect. 4.3 is refined using the ESM method [4]. Finally, it is computed a homography $\mathbf{H^r}$ that maps the unrectified template region to the unrectified query region. Given the homography $\mathbf{H^t}$ that rectifies the template region and the refined homography $\mathbf{H^q}$ that rectifies the query region, then $\mathbf{H^r} = \mathbf{H^q}(\mathbf{H^t})^{-1}$.

5 DARP/DARC selection

This section presents a method for selecting which depth-assisted rectification technique should be used based on the template contents. First, the template image is converted to grayscale. Then a gray-level co-ocurrence matrix (GLCM) [17] is computed from the resulting image. Considering that there are 256 possible gray-levels, the GLCM is a 256×256 matrix. Each entry (i, j) in the matrix contains the number of pixels with gray-level i that are horizontally adjacent to pixels with gray-level j in the image. The GLCM is then normalized by dividing each entry by the sum of all entries. After that, it is calculated the homogeneity property [17] of the obtained GLCM matrix $\mathbf{P}$, which is given by:

$$Homogeneity = \sum_{i,j} \frac{\mathbf{P(i,j)}}{1+|i-j|}. \tag{17}$$

This property ranges between 0 and 1 and measures the concentration of GLCM values in its diagonal. If the homogeneity value is high, it means that most of the horizontally adjacent pixels in the image have close gray-levels, which occurs in texture-less objects. Hence, if the homogeneity metric is low, it is likely that the template image contains a textured object. The selection between DARP and DARC is done by comparing the computed homogeneity property with a threshold (which was set to 0.5): if it is below the threshold, then DARP should be used, otherwise DARC would be the best choice.

6 Using DARP and DARC together

In scenarios where robustness are more critical than performance, one possibility is to use both DARP and DARC methods together in order to detect planar objects. Since the output of both DARP and DARC are point correspondences between query and template images, all these matchings can be used together for detecting objects and computing their pose. Having more matches of different natures can contribute to obtain detection and pose estimation of better quality in some cases. In addition, the simultaneous use of patch and contour features may lead to improved results when dealing with planar objects that have textured and texture-less parts.

However, since every point of a matched contour gives rise to a new correspondence, usually there are much more DARC correspondences than DARP ones, which may lead to a dominance of DARC over DARP. In order to balance the order of magnitude of DARP and DARC correspondence count, one alternative is to perform an uniform sampling over DARC correspondences for keeping a percentage α of them. This way, it is possible to control the probabilities that DARP and

DARC correspondences are chosen by RANSAC when performing pose estimation. The value of the sampling factor α can then be adjusted to change how DARP and DARC contribute to the final result.

7 Frame-to-frame tracking

In a first moment, the DARP and DARC methods described earlier are used for performing tracking by detection, i.e., tracking an object by detecting it at every frame without taking its previous pose into account. Such approach allows automatic initialization and recovery from failures. However, tracking by detection methods are often less accurate/robust than frame-to-frame tracking, where a previous pose estimate is required for computing the current pose of the object. If the object does not move too fast with respect to the camera, its pose on the previous frame can be used as a pose estimate for the current one. This section details how DARP and DARC can also be used for frame-to-frame tracking.

First, tracking is initialized by detecting the object in the previous frame. Then, template features are projected onto the previous frame and, for each feature extracted from the previous frame, it is performed a search for the nearest projected template feature. This is done in order to increase the number of correspondences between the template and the previous frame. Since DARP uses keypoints as features, the metric employed in the search is simply the Euclidean distance between the keypoints locations in the image. For the DARC method, as contours are used, the upright bounding rectangles of the features can be compared. If the distance between a feature from the previous frame and its nearest projected template feature is below a threshold, then a correspondence is established.

Once the correspondences between DARP/DARC features from the previous frame and template features are obtained, the DARP/DARC features from the current frame are matched to previous frame features that have a corresponding template feature using the respective method described in Subsects. 3.6 or 4.4. By transitivity, correspondences are established between current frame and template features. The object pose for the current frame is finally computed from these correspondences using the respective approach detailed in Subsects. 3.6 or 4.4. This process is repeated for the subsequent frames until a tracking failure occurs, when the detection procedure is invoked to reinitialize tracking.

8 Results

This section describes some results obtained with the DARP and DARC methods. The techniques were evaluated regarding performance and pose estimation quality. The hardware used in the evaluations was a Microsoft Kinect for Xbox 360, an Asus Xtion PRO LIVE and a laptop with Intel Core i7-3612QM @ 2.10GHz processor and 8GB RAM. The applications were written in C++ and executed on the Microsoft Windows 7 operating system. The following libraries were used in the implementation of the methods: OpenCV [7], Point Cloud Library (PCL) [51], OpenNI [14] and ESM SDK [4]. The OpenNI library provides ways to compute the intrinsic parameters of the RGB-D sensors from the manufacturer calibration. In addition, it also allows enabling registration between depth and color images, which is performed in the RGB-D sensor hardware. The manufacturer calibration does not consider lens distortion, so one option would be to calibrate the RGB sensor and then use the estimated intrinsic parameters to perform undistortion of registered color and depth images. However, since the RGB-D sensors employed in the evaluations use low distortion lenses [48], the tests did not take into account lens distortion coefficients.

The templates used by DARP and DARC for object detection and pose estimation can be generated with an application where the user interactively draws a rectangle to select the portion of the image where the target object is located, as illustrated in Fig. 12. The user may also provide a binary mask image for determining which image pixels belong to the object to be detected. The DARP method includes all the keypoints within the selected region in the template, while the DARC method uses all the contours inside the selection as a template. DARP templates consist of 2D keypoints (for homography estimation), 3D keypoints (for extrinsic parameters matrix estimation) and keypoint descriptors. DARC templates are composed of 2D contour points, 3D contour points, bounding rectangles of rectified contours and rectifying transformations. If MSER features are used, rectified binary regions and rectifying homographies are additionally stored.

8.1 DARP results

In order to evaluate DARP, the publicly available Technische Universität München's RGBD Datasets [15] were used, which have 1280×960 images. In addition, 320×240 and 640×480 image sequences were captured using the Asus Xtion PRO LIVE and the Microsoft Kinect for Xbox 360 sensors, respectively. Synthetic RGB-D

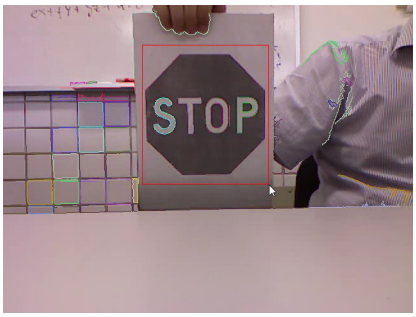


Fig. 12 Template generation application screenshot, where the user selects the object to be detected by drawing a red rectangle around it

images with a resolution of 1280×960 were also generated.

The results obtained when using SIFT [37], ORB [50] and DAFT [15] methods are compared with the results obtained when using these methods together with DARP. Keypoint detection, orientation assignment and patch description are performed in a similar way when each method is used with or without DARP. While SIFT and ORB are based only on RGB data, the DAFT method uses both RGB and depth information.

In the SIFT+DARP scenario, the same algorithms employed by SIFT for keypoint detection, orientation assignment and patch description are used, which are the DoG detector, the gradient orientation histogram method and the SIFT descriptor, respectively [37]. It should be noted that the DoG detector requires an image pyramid for keypoint detection.

In the ORB+DARP scenario, the FAST-9 method is used for keypoint detection [49], but the keypoints are detected on the original scale of the input image, without employing a scale pyramid, since FAST-9 does not use it and scale changes are inherently handled using the patch rectification process. As in ORB, an initial set of features is detected on the input image and then n points with best Harris response are selected. For ORB+DARP it was used a value of $n = 230$ for 640×480 images and $n = 918$ for 1280×960 images in the conducted experiments. ORB uses an image pyramid with 5 levels and a scale factor of 1.2 between consecutive levels in order to obtain scale invariance. When handling 640×480 images, ORB extracts 631 keypoints per image pyramid, distributed in the levels in ascending order as follows: 230, 160, 111, 77 and 53 keypoints. When handling 1280×960 images, ORB extracts 2517 keypoints per image pyramid, distributed in the levels in ascending order as follows: 918, 637, 442, 307 and 213 keypoints. In summary, ORB extracts more keypoints than ORB+DARP, but both approaches handle the same keypoints from the original scale of the input image. ORB and ORB+DARP both use the intensity centroid method for orientation assignment and the rBRIEF patch descriptor [50].

The DAFT+DARP scenario also uses the same methods that DAFT applies for keypoint detection, orientation assignment and patch description, which are a version of the DoG detector that uses depth data [15], Haar wavelet responses orientation histogram [3] and the SURF descriptor [3], respectively. In this case, the keypoint detector needs a depth normalized image pyramid.

Descriptor matching is performed with a nearest neighbor search. For the SIFT and SURF descriptors, a k-d tree is used for obtaining the two nearest neighbors based on the Euclidean distance. Then a heuristic is applied to reject spurious matches, where a correspondence is discarded if the ratio between the distances of the closest and the second-closest neighbor is less than a threshold [37]. In the experiments performed, this threshold was set to 0.7. For the rBRIEF descriptor, a brute force search with Hamming distance was applied, where matches with a distance greater than 50 are discarded. Pose estimation is performed using the same procedures for all the evaluated scenarios, as described in Subsect. 3.6.

8.1.1 Qualitative evaluation

In these experiments, the value of the k parameter for patch size in camera coordinates was empirically set to $\lfloor s/2 \rfloor$, where s is the size of the rectified patch, as mentioned in Subsect. 3.3. Initially the tests were done with planar objects. It is shown in Fig. 13, 14 and Online Resource 1 the matches between two 640×480 images of a planar object. The 2D points that belong to the object model transformed by the homographies computed from the matches are shown in Fig. 15 and Online Resource 1. It can be noted that the ORB+DARP method provides better results than ORB when the object has an oblique pose with respect to the viewing direction. The matches obtained with ORB led to a wrong pose, while it was possible to estimate a reasonable pose using ORB+DARP, as evidenced by the transformed model points (Fig. 15 and Online Resource 1). Scale invariance limit of DARP was also evaluated, as depicted in Fig. 16, 17 and Online Resource 1. It was noted that the DARP method was able to cope with a relative scale change factor of up to 2.5.

After, some tests were done with 640×480 images of non-planar objects with a smooth surface. In this case, Fig. 20 illustrates the projection of a 3D point cloud model of the object using the pose computed from the matches found by ORB+DARP shown in Fig. 19.

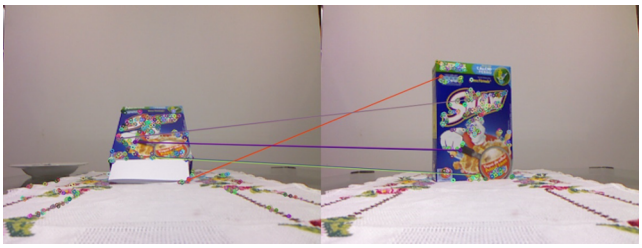


Fig. 13 Planar object keypoint matching using ORB finds 10 matches

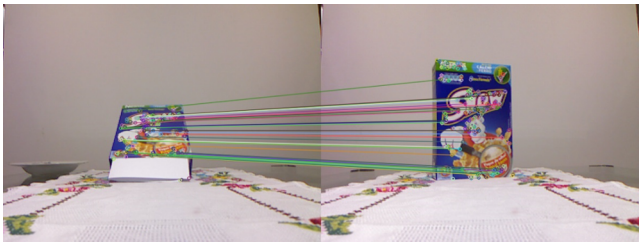


Fig. 14 Planar object keypoint matching using ORB+DARP finds 34 matches

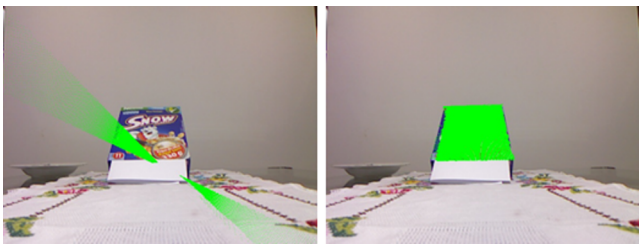


Fig. 15 Planar object pose estimation using ORB (left) and ORB+DARP (right)

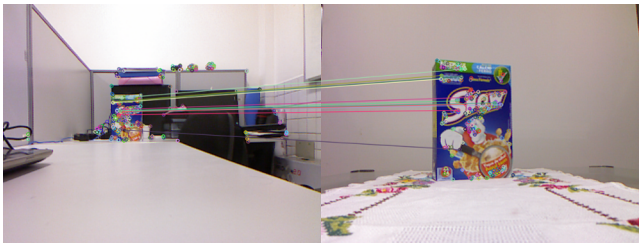


Fig. 16 Scale invariant keypoint matching example using ORB+DARP where 11 matches are found

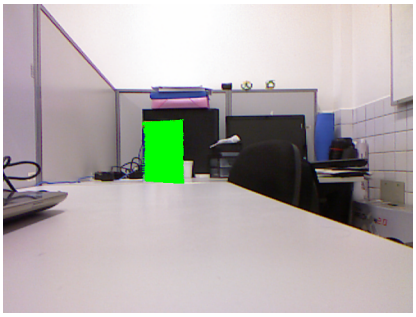


Fig. 17 Scale invariant pose estimation example using ORB+DARP

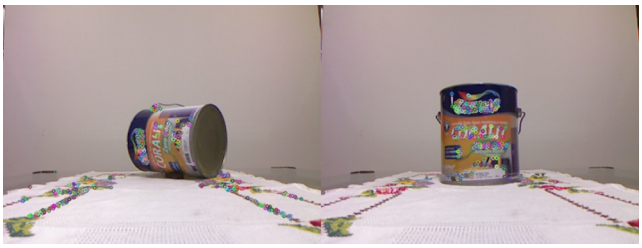


Fig. 18 Non-planar smooth object keypoint matching using ORB finds 0 matches

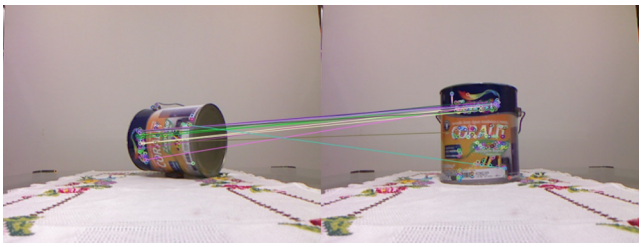


Fig. 19 Non-planar smooth object keypoint matching using ORB+DARP finds 14 matches

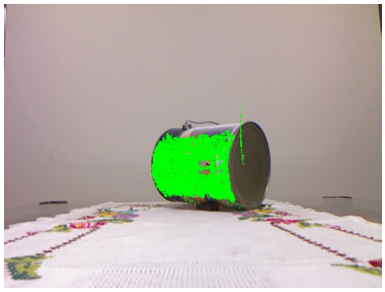


Fig. 20 Non-planar smooth object pose estimation using ORB+DARP

ORB+DARP also obtained better results than ORB in the oblique pose scenario, since ORB+DARP provided matches that allowed computing the object pose, while ORB did not find any valid matches, as can be seen in Fig. 18.

Some experiments were also performed with 320×240 images of non-planar objects with a non-smooth surface. The depth image obtained for such kind of object often contains "holes" caused by inter-occlusions between parts of the object, as can be seen in Fig. 21 left. In order to obtain better results, the template depth image was enhanced with the help of Kinect Fusion [44]. In order to do this, it was needed to capture a sequence of depth images of the object taken from different views. The resulting depth image is illustrated in Fig. 21 right.

In some cases, such as the one depicted in Fig. 22 and 23, ORB+DARP is able to correctly perform keypoint matching and pose estimation in the non-planar non-smooth surface scenario. Nevertheless, there are times where ORB succeeds (Fig. 24 and 26 left) and

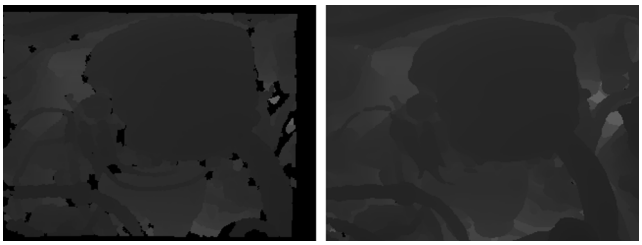


Fig. 21 Original depth map (left) and depth map obtained using Kinect Fusion (right)

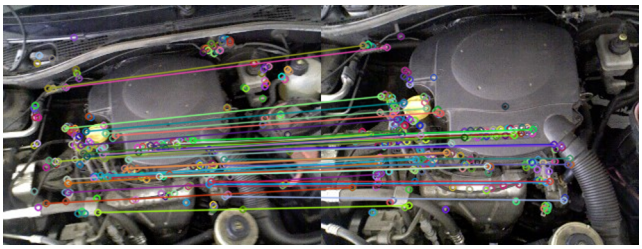


Fig. 22 Success case of non-planar non-smooth object keypoint matching using ORB+DARP, where 42 matches are found

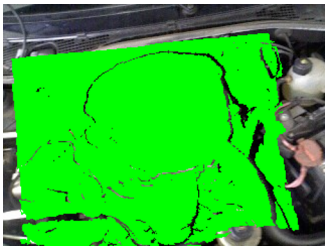


Fig. 23 Success case of non-planar non-smooth object pose estimation using ORB+DARP

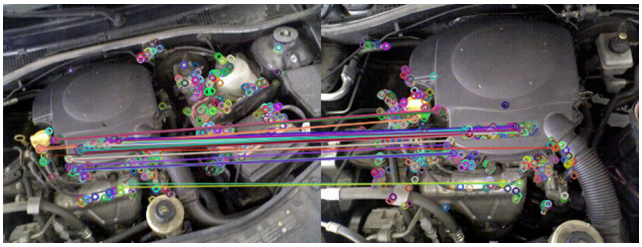


Fig. 24 Success case of non-planar non-smooth object keypoint matching using ORB, where 47 matches are found

ORB+DARP fails (Fig. 25 and 26 right) when dealing with non-planar non-smooth objects. This can be explained by the fact that non-smooth objects may not have well defined normals along their entire surface, which may harm patch rectification.

8.1.2 Quantitative evaluation

Keypoint matching quality was evaluated by measuring the correctness of the poses estimated from the matches. The first evaluation was done with a database of 2560 synthetic RGB-D images of a planar object (a

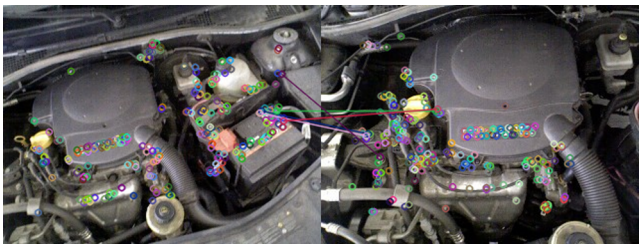


Fig. 25 Failure case of non-planar non-smooth object keypoint matching using ORB+DARP, where 5 matches are found

Fig. 26 Non-planar non-smooth object pose estimation is successful when ORB is used (left), while it fails when ORB+DARP is used (right)

cereal box) under different viewpoints on a cluttered background. Some frames from the generated synthetic dataset are depicted in Fig. 27. In order to generate these images, the object was placed on the origin of a spherical coordinate system whose equatorial plane coincides with the xz plane of the object coordinate system, as illustrated in Fig. 28. The camera always looks at the origin of the coordinate system and a pose can be defined by a latitude ϕ, a longitude λ, a camera roll ω and a distance d to the origin (which relates to object scale). When generating the dataset, viewpoints with a given degree change θ are obtained by considering 8 different (ϕ, λ) combinations: $(-\theta, -\theta)$, $(-\theta, 0)$, $(-\theta, \theta)$, $(0, -\theta)$, $(0, \theta)$, $(\theta, -\theta)$, $(\theta, 0)$ and (θ, θ). The poses were under a degree change range of $[10\,°, 80\,°]$ with a $10\,°$ step, a camera roll range of $[0\,°, 360\,°]$ with a $45\,°$ step and a scale range of $[1.0, 1.8]$ with a 0.2 step. Summing up, 8 different degree changes (each one with 8 combinations of ϕ and λ), 8 different camera roll angles and 5 different scales were used, totalizing 2560 different poses.

As in [25], the metric used in the evaluation was the percentage of correct poses estimated by each method. In many works (e.g. [55]) it is considered that a correspondence is an inlier when its reprojection error is less than 3 pixels. Due to this, a pose was considered as correct only if the root-mean-square (RMS) reprojection error was below 3 pixels. The k parameter was the same used in the qualitative evaluation. In larger viewpoint changes it can be seen that SIFT+DARP,

Fig. 27 Images from the cereal box synthetic RGB-D dataset, where the viewpoint change is shown below the respective image

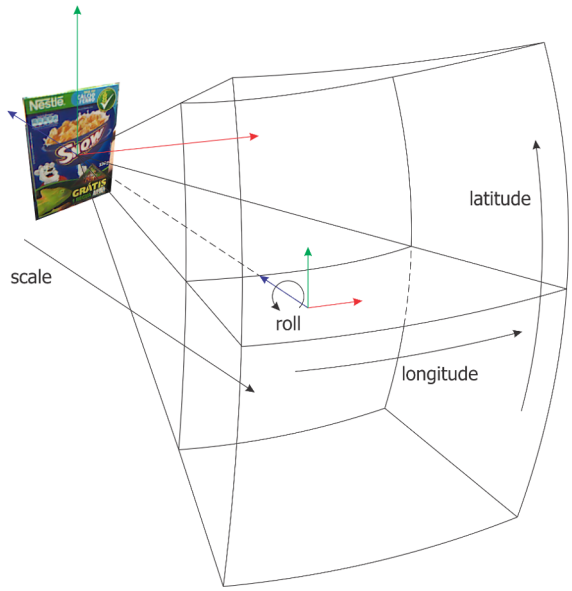
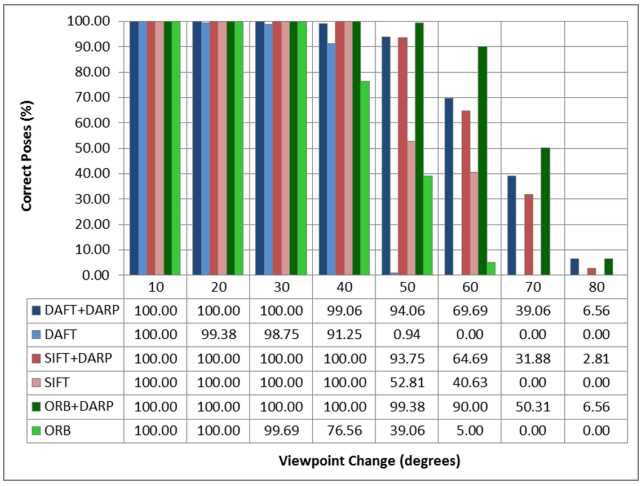


Fig. 28 Spherical coordinate system used for generating the synthetic dataset



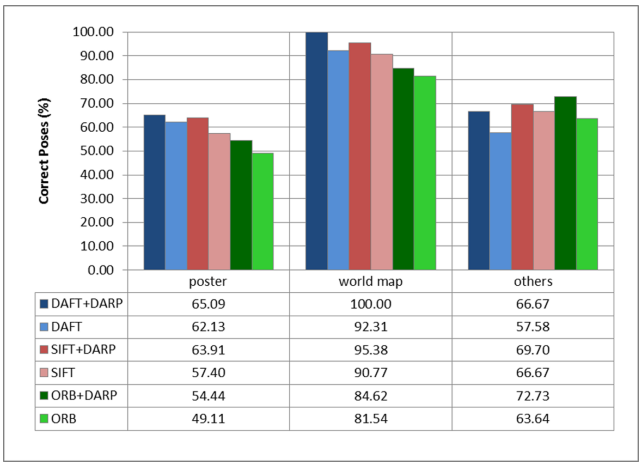
	10	20	30	40	50	60	70	80
DAFT+DARP	100.00	100.00	100.00	99.06	94.06	69.69	39.06	6.56
DAFT	100.00	99.38	98.75	91.25	0.94	0.00	0.00	0.00
SIFT+DARP	100.00	100.00	100.00	100.00	93.75	64.69	31.88	2.81
SIFT	100.00	100.00	100.00	100.00	52.81	40.63	0.00	0.00
ORB+DARP	100.00	100.00	100.00	100.00	99.38	90.00	50.31	6.56
ORB	100.00	100.00	99.69	76.56	39.06	5.00	0.00	0.00

Fig. 29 Percentage of correct poses with respect to viewpoint change of the evaluated approaches with the cereal box synthetic RGB-D database

DAFT+DARP and ORB+DARP outperformed SIFT, DAFT and ORB, respectively, as shown in Fig. 29.

The RGB-D datasets from Technische Universität München [15] were also used to quantitatively evaluate the different methods regarding pose estimation quality. Some frames from these datasets are shown in Fig. 30.

The *poster* and *world map* datasets were used in separate, since they have several images under different rotations, scales and viewpoints. The remaining datasets (*frosties* and *granada*), which have fewer images, were evaluated all together under the label *others*. In these experiments, the k parameter was empirically set to $((d/f)+1)\lfloor s/2 \rfloor$, where d is the average distance between the target object and the camera (which was set



	poster	world map	others
DAFT+DARP	65.09	100.00	66.67
DAFT	62.13	92.31	57.58
SIFT+DARP	63.91	95.38	69.70
SIFT	57.40	90.77	66.67
ORB+DARP	54.44	84.62	72.73
ORB	49.11	81.54	63.64

Fig. 31 Percentage of correct poses with respect to viewpoint change of the evaluated approaches with The Technische Universität München's RGBD Datasets [15]

to 2 meters), f is the focal length and s is the size of the rectified patch (see Subsect. 3.3). Fig. 31 shows that results obtained with SIFT+DARP, DAFT+DARP and ORB+DARP are better than the ones obtained with SIFT, DAFT and ORB, respectively.

8.1.3 Performance analysis

RGB-D images with a resolution of 640×480 pixels were used to analyze the performance of a non-optimized version of the DARP method. Table 1 presents the average time and the percentage of time required by each step of ORB and ORB+DARP, which are the fastest approaches among the ones that were evaluated. It shows that the ORB+DARP method runs at ∼29 fps and its most time demanding step is the normal estimation phase, which takes almost 50% of all processing time. The patch rectification step also heavily contributes to the final processing time. ORB takes more time than ORB+DARP for keypoint detection and patch description, since it uses an image pyramid and extracts a higher number of keypoints. ORB estimates patch orientation in a faster manner than ORB+DARP because it makes use of integral images in this step. ORB+DARP could be optimized to perform orientation estimation in the same way, but it would not represent a significant performance gain, as this step takes less than 1% of total processing time.

8.2 DARC results

To the best of the authors' knowledge, there is no publicly available RGB-D image dataset of texture-less planar objects. Due to that, synthetic RGB-D images of

Fig. 30 Images from the Technische Universität München's RGBD Datasets [15], where the dataset name is shown below the respective image

Table 1 Average computation time and percentage for each step of ORB and ORB+DARP methods when handling 640 × 480 RGB-D images

	ORB		ORB+DARP	
	ms	**%**	**ms**	**%**
Keypoint detection	21.90	80.63	4.96	14.25
Normal estimation	–	–	17.24	49.52
Patch rectification	–	–	9.64	27.69
Orientation estimation	0.14	0.53	0.18	0.51
Patch description	5.12	18.84	2.80	8.03
Total	**27.16**	**100.00**	**34.82**	**100.00**

texture-less objects with a resolution of 1280×960 were generated in order to evaluate DARC. In addition, some image sequences were captured using the Microsoft Kinect for Xbox 360.

8.2.1 Qualitative evaluation

Fig. 32 and Online Resource 2 show some results obtained with DARC for detection and pose estimation of different planar objects. It can be seen that DARC can deal with significant changes in rotation and scale as well as with perspective distortions. The contour groups used as templates are the octagon of the stop sign together with its inner contours, the continent frontier of the map and the outer square of the logo together with its inner contours.

Similarly to [30], the use of depth information allows DARC to distinguish objects that have the same shape but different sizes, as illustrated in Fig. 33 and Online Resource 2. The virtual objects are rendered with a different color and size depending on the size of the detected object. Detection methods that are based solely on RGB data are not able to differentiate, for example, between a small object at a close distance and a big object at a far distance when their projections have the same shape and size. DARC is also capable of detecting objects even when they are partially occluded, as shown in Fig. 34 and Online Resource 2, and is able to handle a relative scale change factor of up to 5.0, as illustrated in Fig. 35 and Online Resource 2.

8.2.2 Quantitative evaluation

DARC was compared to some existing techniques regarding pose estimation quality and performance. Three texture based techniques were selected for the evaluation: SIFT, ORB and DAFT. The algorithms used by each method for keypoint matching and pose estimation are described in Subsect. 8.1. It should be noted that DAFT also uses both RGB and depth images, as well as DARC. In addition, the Perspective Template Matching

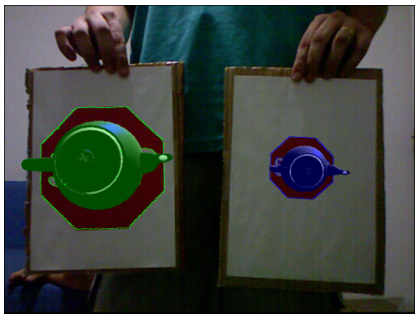


Fig. 33 Distinction of objects with the same shape and different sizes using DARC. The bigger stop sign is augmented with a bigger green teapot, while the smaller stop sign is augmented with a smaller blue teapot

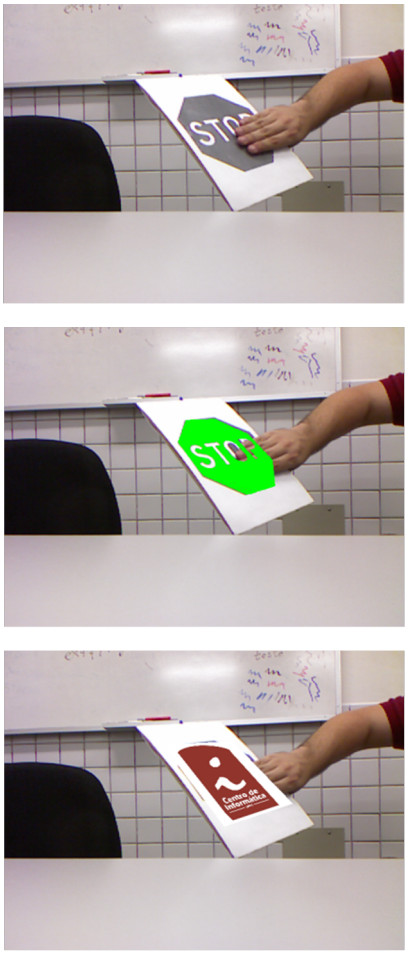


Fig. 34 Occlusion handling using DARC: input image (top), detection result (middle) and augmentation (bottom)

(PTM) technique [24], which exploits contour information, is also evaluated. It makes use of deformable edge templates together with a coarse-to-fine search in order to detect texture-less planar objects.

Two different configurations of the DARC method were compared: DARC-CC, which uses the Canny contour detector and the chamfer matcher; and DARC-

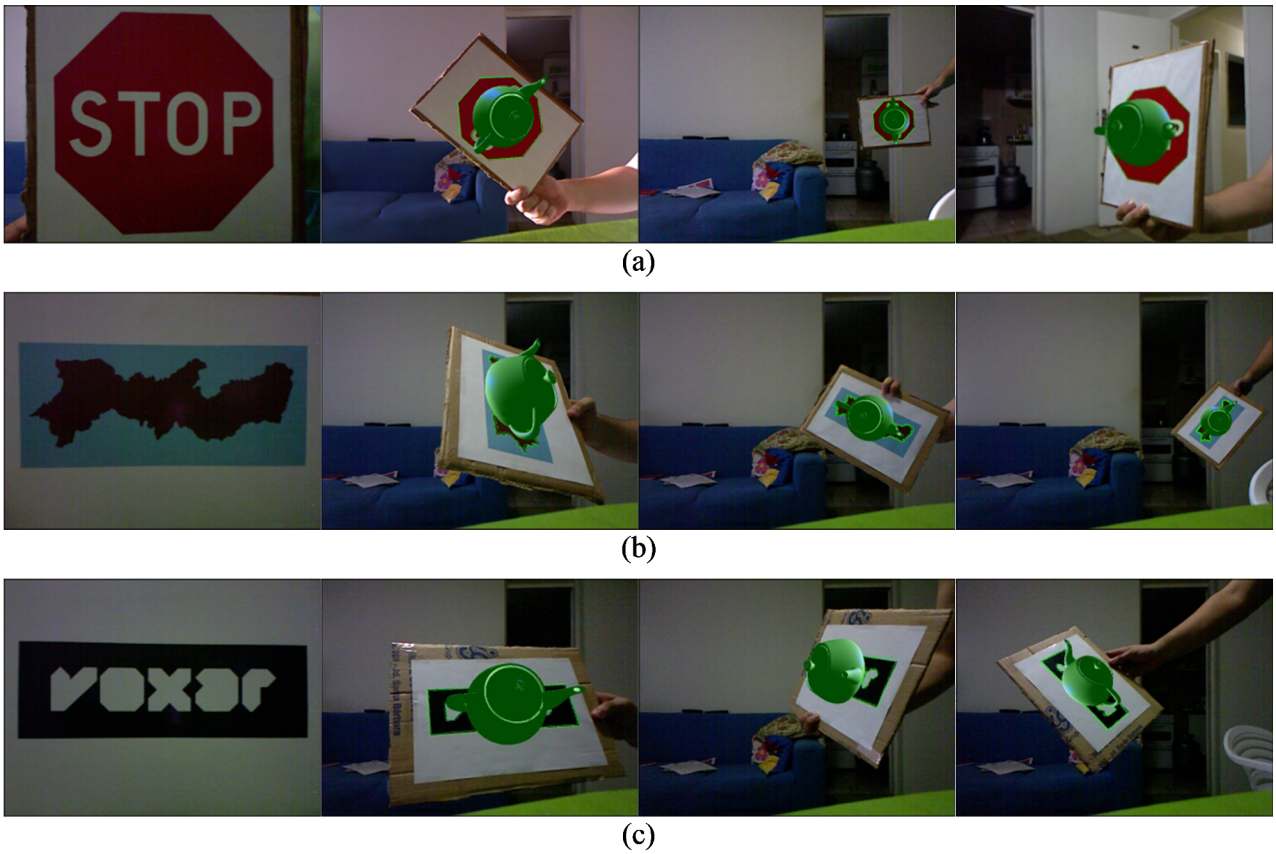


Fig. 32 Augmentation of planar objects under different poses using DARC. The proposed method is used to augment a traffic sign (a), a map (b) and a logo (c). The leftmost image of each group shows the object to be detected

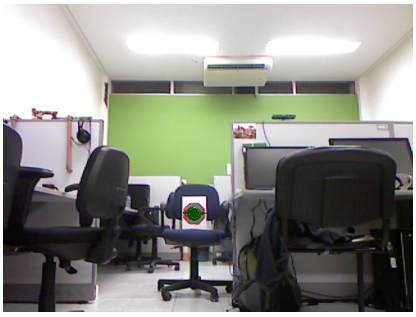
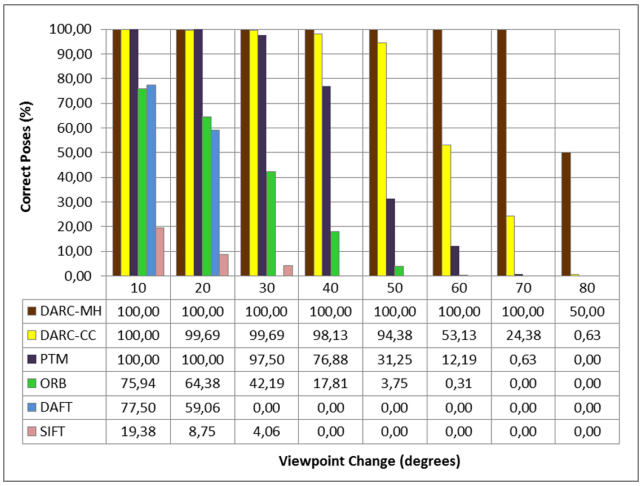


Fig. 35 Scale invariant pose estimation of a stop sign using DARC



	10	20	30	40	50	60	70	80
DARC-MH	100,00	100,00	100,00	100,00	100,00	100,00	100,00	50,00
DARC-CC	100,00	99,69	99,69	98,13	94,38	53,13	24,38	0,63
PTM	100,00	100,00	97,50	76,88	31,25	12,19	0,63	0,00
ORB	75,94	64,38	42,19	17,81	3,75	0,31	0,00	0,00
DAFT	77,50	59,06	0,00	0,00	0,00	0,00	0,00	0,00
SIFT	19,38	8,75	4,06	0,00	0,00	0,00	0,00	0,00

Fig. 37 Percentage of correct poses with respect to viewpoint change of the evaluated approaches with the stop sign synthetic RGB-D database

MH, which uses the MSER contour detector and the Hamming matcher.

Pose estimation quality was evaluated with a synthetic database of 2560 RGB-D images of a stop sign under different viewpoints on a cluttered background. Some frames from this dataset are shown in Fig. 36. The contour group that contains the octagon of the stop sign together with its inner contours was used as template. The pose range and the metric for considering a pose as correct were the same used in the evaluation with a synthetic dataset described in Subsect. 8.1. As can be noted in Fig. 37, DARC outperformed all the other methods in all larger viewpoint changes. It can also be noted that DARC-MH provided better results than DARC-CC.

Fig. 36 Images from the stop sign synthetic RGB-D dataset, where the viewpoint change is shown below the respective image

8.2.3 Performance analysis

In the experiments presented in this subsection it was used the same stop sign template as described in the previous subsection. The fastest method for keypoint matching among the ones evaluated is ORB, and its performance when dealing with 640 × 480 RGB-D images was already presented in Subsect. 8.1. In the same scenario the PTM technique takes more than one second to detect a template. The performance of each step of non-optimized implementations of DARC-CC and DARC-MH when detecting a single contour group in 640 × 480 RGB-D images is compared in Table 2. Distance transform is only performed by DARC-CC. It is shown that DARC-CC runs at ∼36 fps and DARC-MH runs at ∼15 fps while detecting a single contour group. If most of the contour groups in the scene do not have a size similar to any template contour group size, they are quickly discarded by DARC, not affecting the application performance. Due to this, DARC frame rate is more influenced by the number of detected template contour groups on the scene than by the number of template contour groups in the database. This metric was taken into account on the following experiments. Regarding the other methods evaluated in the previous subsection, PTM performance is also directly influenced by the number of detected templates, while the performance of keypoint matching methods such as ORB, SIFT and DAFT is not much affected by this factor.

Table 2 Average computation time and percentage for each step of DARC-CC and DARC-MH methods when handling 640 × 480 RGB-D images

	DARC-CC		**DARC-MH**	
	ms	**%**	**ms**	**%**
Contour detection	6.18	22.38	42.05	64.71
Distance transform	7.16	25.92	–	–
Normal and orientation estimation	0.25	0.90	2.68	4.14
Contour rectification	0.54	1.96	12.74	19.61
Contour matching	1.40	5.05	6.29	9.68
Coarse pose refinement	12.10	43.79	1.21	1.86
Total	**27.63**	**100.00**	**64.97**	**100.00**

The average time and percentage of time required by each step of DARC-CC for different amounts of detected templates are depicted in Fig. 38 and 39, respectively. For DARC-CC, the bottlenecks are contour detection, distance transform and coarse pose refinement, which take together more than 90% of all processing time when detecting a single template. However, it should be noted that the contour detection and the distance transform times are relatively constant, while

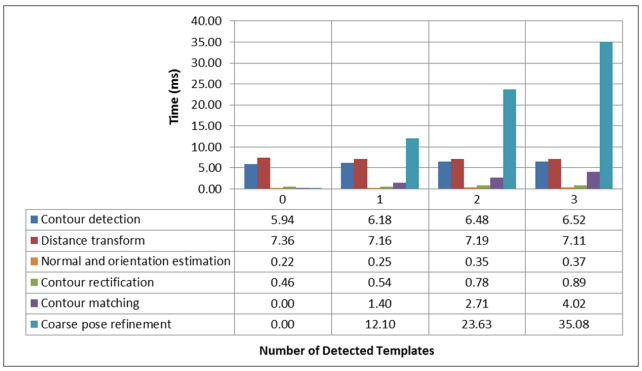


Fig. 38 Average computation time of each step of DARC-CC for different numbers of detected templates

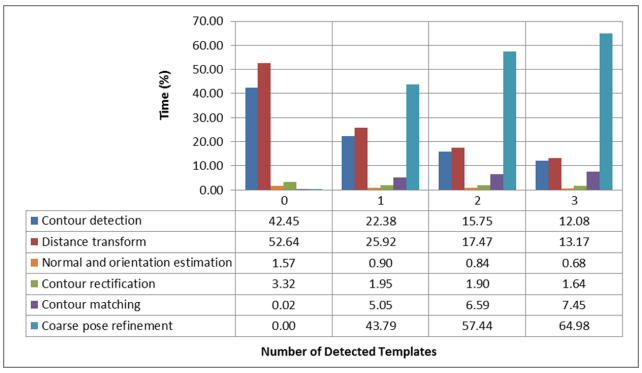


Fig. 39 Percentage of time of each step of DARC-CC for different numbers of detected templates

the coarse pose refinement time grows linearly with the number of detected templates.

The average time and percentage of time required by each step of DARC-MH for different amounts of detected templates are shown in Fig. 40 and 41, respectively. For DARC-MH, the major bottleneck is contour detection, since it takes alone almost 65% of all processing time when detecting a single template, but its time remains relatively constant. It can also be noted that contour matching and coarse pose refinement times in DARC-MH grow linearly with respect to the number of detected templates.

8.3 DARP/DARC selection results

In order to evaluate the DARP/DARC selection method, it was used some of the tracking targets available in the Metaio template-based tracking dataset [32]. The tracking targets in this dataset are classified regarding their texturedness, so it was possible to check if the proposed selection method provided results that are consistent with this classification. In the performed experiments, the ORB+DARP and DARC-MH variants were used, since they offer a good trade-off between

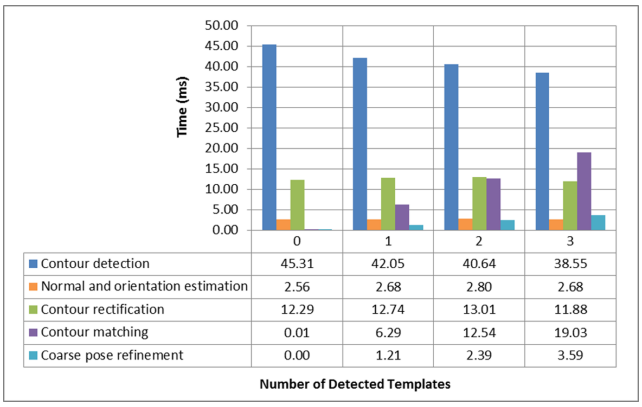


Fig. 40 Average computation time of each step of DARC-MH for different numbers of detected templates

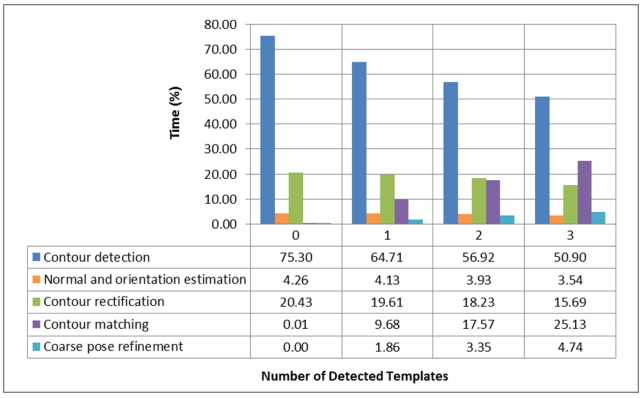


Fig. 41 Percentage of time of each step of DARC-MH for different numbers of detected templates

pose estimation quality and performance, as shown in previous subsections. Due to the scalability issue regarding the number of contour groups of DARC-MH reported in the previous subsection, DARC-MH template was limited to contain the five contour groups with largest area. Fig. 42 presents the results obtained with the proposed selection method using two texture-less objects (*bump* and *stop*) and two textured objects (*lucent* and *philadelphia*) from the Metaio dataset. It can be noted that the GLCM homogeneity property computed from each template image is consistent with the classification performed in the Metaio dataset. Fig. 42 also compares augmentation results obtained using ORB+DARP and DARC-MH for a given query frame of each template. While ORB+DARP succesfully augments the objects that were classified as textured by the proposed method, it fails when handling the texture-less ones. With DARC-MH, the opposite occurs: it is able to correctly detect and augment the objects that are texture-less according to the homogeneity property and it fails to detect the textured objects. These results indicate that the proposed method could be used

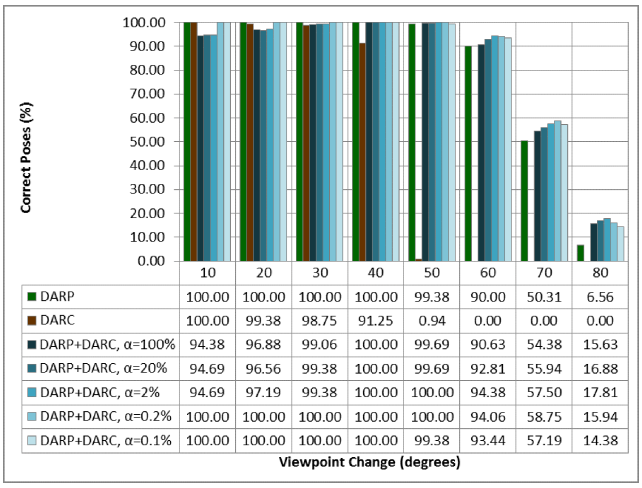


Fig. 43 Percentage of correct poses with respect to viewpoint change of DARP, DARC and DARP+DARC with the cereal box synthetic RGB-D database

to choose between DARP or DARC based on the template contents.

8.4 DARP+DARC results

In the evaluation of the combined use of DARP and DARC, the ORB+DARP and DARC-MH variants were employed again and are referred here simply as DARP and DARC, respectively. As described in Subsect. 8.3, the DARC template consisted only of the five contour groups with largest area. Pose estimation quality was evaluated using only DARP, only DARC and using both DARP and DARC together (DARP+DARC). Different values for the sampling factor α were also tested. In the performed experiments, typical values for the number of point correspondences were around 500 for DARP and around $50,000$ for DARC (without sampling).

First, it was performed an evaluation using the cereal box dataset described in Subsect. 8.1. The obtained results are illustrated in Fig. 43. It can be seen that using DARP alone was better than using solely DARC, which was expected since the target is textured. It can also be noted that DARP+DARC without sampling ($\alpha = 100\%$) outperformed DARP in the cases with higher viewpoint change, but DARP was better in the situations with lower viewpoint change. However, it is shown that decreasing the sampling factor α until a certain limit results in an improvement in DARP+DARC that makes it outperform DARP in all viewpoint changes. The experiments showed that a value of 0.2% for α provided good results.

In order to analyze the combination of DARP and DARC when dealing with a planar object that has textured and texture-less parts, it was generated a syn-

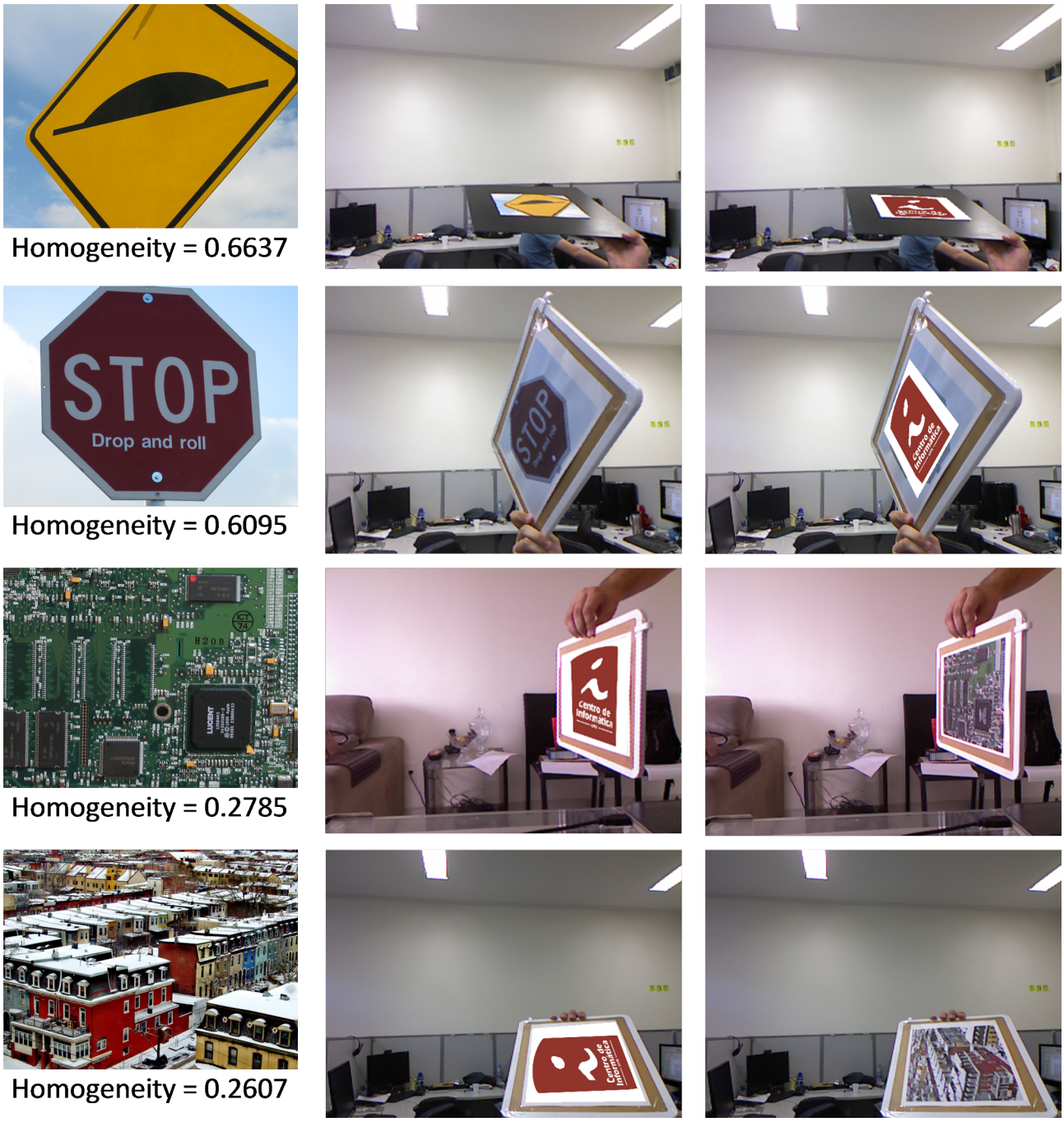
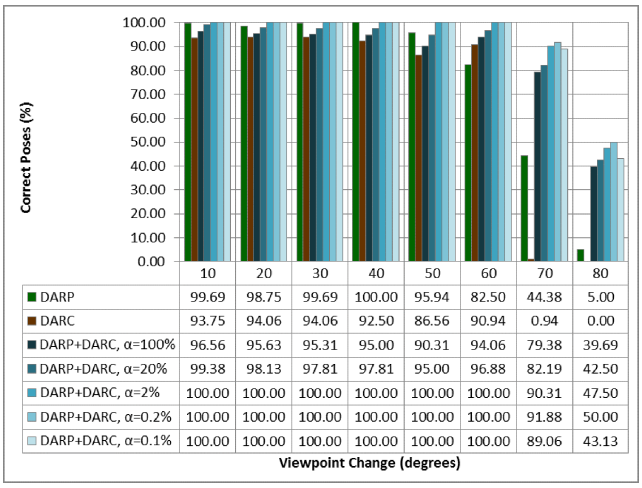


Fig. 42 Depth-assisted rectification method selection results. First column: texture-less (two top rows) and textured (two bottom rows) template images and their respective GLCM homogeneity. Second column: augmentation results using ORB+DARP. Third column: augmentation results using DARC-MH

thetic dataset in a similar manner to what is described in Subsect. 8.1 but using an object composed of the cereal box from Subsect. 8.1 side by side to the stop sign from Subsect. 8.2. Fig. 44 shows the evaluation results. Due to the texture-less part, using DARC alone when dealing with higher viewpoint changes provided better results than in the previous experiment. However, it was worse than using solely DARP in most viewpoint changes. It can also be seen that DARP+DARC exhibited the same behaviour than in the previous experiment, in a way that it outperformed DARP in all viewpoint changes when using a value of α such as 0.2%.



	10	20	30	40	50	60	70	80
DARP	99.69	98.75	99.69	100.00	95.94	82.50	44.38	5.00
DARC	93.75	94.06	94.06	92.50	86.56	90.94	0.94	0.00
DARP+DARC, α=100%	96.56	95.63	95.31	95.00	90.31	94.06	79.38	39.69
DARP+DARC, α=20%	99.38	98.13	97.81	97.81	95.00	96.88	82.19	42.50
DARP+DARC, α=2%	100.00	100.00	100.00	100.00	100.00	100.00	90.31	47.50
DARP+DARC, α=0.2%	100.00	100.00	100.00	100.00	100.00	100.00	91.88	50.00
DARP+DARC, α=0.1%	100.00	100.00	100.00	100.00	100.00	100.00	89.06	43.13

Fig. 44 Percentage of correct poses with respect to viewpoint change of DARP, DARC and DARP+DARC with the cereal box+stop sign synthetic RGB-D database

8.5 Frame-to-frame tracking results

A number of frames from the *cereal_box_1* sequence of the University of Washington's RGB-D Object Dataset [29] were used to evaluate the proposed frame-to-frame tracking approach, since the pose change between consecutive frames in this dataset is small. Since the object in the sequence is textured, the ORB+DARP variant was adopted. Fig. 45 depicts the results of both tracking by detection and frame-to-frame tracking in this scenario. While the tracking by detection approach fails in the last frames of the sequence, the frame-to-frame procedure is able to track the object throughout the whole sequence. The procedure for increasing the number of correspondences between the previous frame and the template took less than 1.5 ms per frame.

9 Conclusions

It was shown that the use of RGB-D sensors allows improving object detection and tracking from natural features. The DARP method has been proposed, which exploits depth information to improve keypoint matching. This is done by rectifying the patches using the 3D information in order to remove perspective effects. The depth information is also used to obtain a scale invariant representation of the patches. It was shown that DARP can be used together with existing keypoint matching methods in order to help them to handle situations such as oblique poses with respect to the viewing direction. It supports both planar and non-planar objects and is able to run in real time. The DARC technique has also been proposed, which performs detection and pose estimation of texture-less planar objects by making use of depth information available in RGB-D consumer devices. In order to achieve this, contours extracted from a query image are rectified for removing distortions caused by rotation, scale and perspective transforms. The normalized representation is matched to templates acquired a priori and a coarse pose is calculated, which is then refined using optimization methods. DARC showed to be robust to in-plane and out-of-plane rotations, scale and perspective deformations, being able to compute the pose of planar objects in real-time. DARC-MH showed to be more robust and accurate but slower than DARC-CC. The choice of what is the best DARC setup is application dependent: if robustness is more crucial than performance, DARC-MH should be preferred; otherwise, DARC-CC is the best option. It was also shown that it is possible to choose the most suitable method between DARP and DARC for detecting a given object by checking the GLCM homogeneity property of an object's image. When dealing with planar textured objects or planar objects that have textured and texture-less parts, it was shown that using both DARP and DARC together may lead to a more robust detection, considering that the performance constraints are not so hard. In addition, it was demonstrated that depth-assisted rectification can be used for both tracking by detection and frame-to-frame tracking, taking benefit from both worlds: automatic initialization and recovery from failures of tracking by detection and accuracy/robustness of frame-to-frame tracking.

Current limitations of the DARP method are: it requires additional time for rectifying features, especially due to the normal estimation step; it still fails in some extreme cases of severe perspective or scale distortions; and it may fail when handling some non-planar objects with a non-smooth surface. DARC limitations that should be mentioned are: it does not target non-planar objects; its computational performance drops linearly with the number of detected templates; and it fails in some scenarios where the contours do not have a shape that is discriminative enough or extremely severe perspective/scale distortions occur.

As future work regarding DARP, it will be evaluated how normal estimation can be speeded up, maybe using faster approaches such as the one described in [21]. An implementation on GPU may also be used for optimization purposes. The effect of using a few image pyramid levels and different patch sizes in camera coordinates instead of a single level and patch size will also be evaluated. It will be studied if it is possible to determine automatically the optimal patch size in camera coordinates for a given scene. A refinement step for patch pose estimation using a template track-

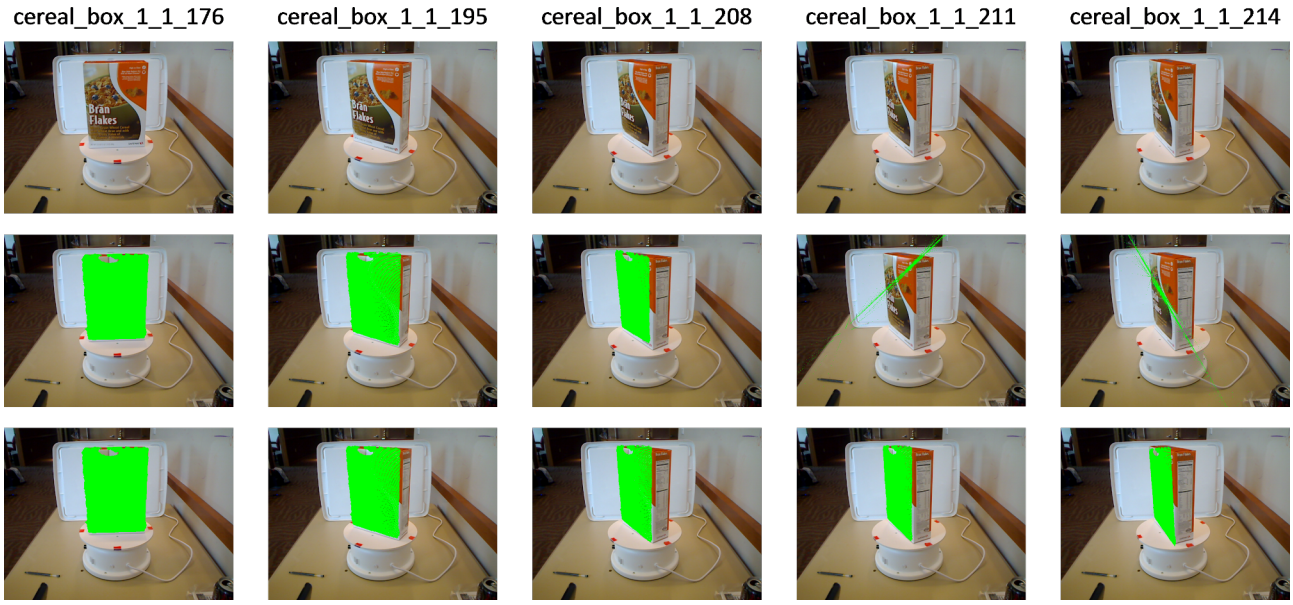


Fig. 45 Comparison of tracking by detection and frame-to-frame tracking using ORB+DARP. First row: sequence frames examples from the University of Washington's RGB-D Object Dataset [29] and their respective names. Second row: results obtained using tracking by detection. Third row: results obtained using frame-to-frame tracking

ing method such as [4] will be considered. Another issue that should be investigated is that when the object suffers from severe perspective or scale distortion, the rectified patch loses resolution, which impacts on its description. One alternative to be studied for solving this would be to generate distorted versions of the reference images prior to keypoint matching [8]. Then, the available depth and normal information could be used to select a set of most probable matching keypoints for each patch. DARP support for non-planar non-smooth objects should also be improved, perhaps by obtaining a parameterization of the 3D surface that would allow flattening the non-planar object for obtaining a planar representation of it. This would use an approach similar to the one described in [43], where B-splines surfaces are fitted to point clouds obtained from RGB-D sensors. With respect to DARC, GPU optimization should also be considered. It will be evaluated the possibility of extending the technique for working with non-planar objects. A verification method using neighboring contours such as the one described in [25] could also be used. Confusions can occur when the template contour groups do not have enough discriminative power. It will be studied if the discriminative power of contour matching can be improved by making use of oriented chamfer matching [52] or directional chamfer matching [36].

Acknowledgements The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) / Institut National de Recherche en Informatique et en Automatique (INRIA) / Comisión Nacional de Investigación Científica y Tecnológica (CONICYT) STIC-AmSud project ARVS and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) (process 141705/2010-8) for partially funding this research.

References

1. Álvarez, H., Borro, D.: Junction assisted 3d pose retrieval of untextured 3d models in monocular images. Computer Vision and Image Understanding **117**(10), 1204–1214 (2013)
2. Barrow, H.G., Tenenbaum, J.M., Bolles, R.C., Wolf, H.C.: Parametric correspondence and chamfer matching: Two new techniques for image matching. Tech. rep., DTIC Document (1977)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer vision and image understanding **110**(3), 346–359 (2008)
4. Benhimane, S., Malis, E.: Real-time image-based tracking of planes using efficient second-order minimization. In: Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, vol. 1, pp. 943–948. IEEE (2004)
5. Berkmann, J., Caelli, T.: Computation of surface geometry and segmentation using covariance techniques. Pattern Analysis and Machine Intelligence, IEEE Transactions on **16**(11), 1114–1116 (1994)
6. Borgefors, G.: Distance transformations in digital images. Computer vision, graphics, and image processing **34**(3), 344–371 (1986)

7. Bradski, G., Kaehler, A.: Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc. (2008)
8. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: Computer Vision–ECCV 2010, pp. 778–792. Springer (2010)
9. Canny, J.: A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on (6), 679–698 (1986)
10. Cruz, L., Lucio, D., Velho, L.: Kinect and rgbd images: Challenges and applications. In: Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on, pp. 36–49. IEEE (2012)
11. Del Bimbo, A., Franco, F., Pernici, F.: Local homography estimation using keypoint descriptors. In: Analysis, Retrieval and Delivery of Multimedia Content, pp. 203–217. Springer (2013)
12. Donoser, M., Kontschieder, P., Bischof, H.: Robust planar target tracking and pose estimation from a single concavity. In: Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on, pp. 9–15. IEEE (2011)
13. Eyjolfsdottir, E., Turk, M.: Multisensory embedded pose estimation. In: Applications of Computer Vision (WACV), 2011 IEEE Workshop on, pp. 23–30. IEEE (2011)
14. Falahati, S.: OpenNI Cookbook. Packt Publishing Ltd (2013)
15. Gossow, D., Weikersdorfer, D., Beetz, M.: Distinctive texture features from perspective-invariant keypoints. In: Pattern Recognition (ICPR), 2012 21st International Conference on, pp. 2764–2767. IEEE (2012)
16. Hagbi, N., Bergig, O., El-Sana, J., Billinghurst, M.: Shape recognition and pose estimation for mobile augmented reality. In: Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on, pp. 65–71. IEEE (2009)
17. Haralick, R.M.: Lg shapiro. computer and robot vision, vol. 1 (1992)
18. Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey vision conference, vol. 15, p. 50. Manchester, UK (1988)
19. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
20. Hinterstoisser, S., Benhimane, S., Navab, N., Fua, P., Lepetit, V.: Online learning of patch perspective rectification for efficient object detection. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. IEEE (2008)
21. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 858–865. IEEE (2011)
22. Hinterstoisser, S., Kutter, O., Navab, N., Fua, P., Lepetit, V.: Real-time learning of accurate patch rectification. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 2945–2952. IEEE (2009)
23. Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., Navab, N.: Dominant orientation templates for real-time detection of texture-less objects. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2257–2264. IEEE (2010)
24. Hofhauser, A., Steger, C., Navab, N.: Edge-based template matching and tracking for perspectively distorted planar objects. In: Advances in Visual Computing, pp. 35–44. Springer (2008)
25. Holzer, S., Hinterstoisser, S., Ilic, S., Navab, N.: Distance transform templates for object detection and pose estimation. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 1177–1184. IEEE (2009)
26. Konolige, K.: Projected texture stereo. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, pp. 148–155. IEEE (2010)
27. Koser, K., Koch, R.: Perspectively invariant normal features. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1–8. IEEE (2007)
28. Kurz, D., Benhimane, S.: Gravity-aware handheld augmented reality. In: Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on, pp. 111–120. IEEE (2011)
29. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 1817–1824. IEEE (2011)
30. Lee, W., Park, N., Woo, W.: Depth-assisted real-time 3d object detection for augmented reality. ICAT'11 pp. 126–132 (2011)
31. Lepetit, V., Lagger, P., Fua, P.: Randomized trees for real-time keypoint recognition. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2, pp. 775–781. IEEE (2005)
32. Lieberknecht, S., Benhimane, S., Meier, P., Navab, N.: A dataset and evaluation methodology for

template-based tracking algorithms. In: ISMAR, pp. 145–151 (2009)
33. Lima, J.P., Simoes, F., Uchiyama, H., Teichrieb, V., Marchand, E., et al.: Depth-assisted rectification of patches using rgb-d consumer devices to improve real-time keypoint matching. In: Int. Conf. on Computer Vision Theory and Applications, Visapp 2013, pp. 651–656 (2013)
34. Lima, J.P., Teichrieb, V., Uchiyama, H., Marchand, E., et al.: Object detection and pose estimation from natural features using consumer rgb-d sensors: Applications in augmented reality. In: IEEE Int. Symp. on Mixed and Augmented Reality (doctoral symposium), ISMAR'12, pp. 1–4 (2012)
35. Lima, J.P., Uchiyama, H., Teichrieb, V., Marchand, E.: Texture-less planar object detection and pose estimation using depth-assisted rectification of contours. In: Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on, pp. 297–298. IEEE (2012)
36. Liu, M.Y., Tuzel, O., Veeraraghavan, A., Chellappa, R.: Fast directional chamfer matching. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 1696–1703. IEEE (2010)
37. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
38. Marcon, M., Frigerio, E., Sarti, A., Tubaro, S.: 3d wide baseline correspondences using depth-maps. Signal Processing: Image Communication **27**(8), 849–855 (2012)
39. Martedi, S., Thomas, B., Saito, H.: Region-based tracking using sequences of relevance measures. In: Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on, pp. 1–6. IEEE (2013)
40. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of the British Machine Vision Conference, vol. 1, pp. 384–393. BMVA (2002)
41. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. International journal of computer vision **65**(1-2), 43–72 (2005)
42. Morel, J.M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences **2**(2), 438–469 (2009)
43. Morwald, T., Richtsfeld, A., Prankl, J., Zillich, M., Vincze, M.: Geometric data abstraction using b-splines for range image segmentation. In: Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 148–153. IEEE (2013)
44. Newcombe, R.A., Davison, A.J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on, pp. 127–136. IEEE (2011)
45. Ozuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pp. 1–8. Ieee (2007)
46. Pagani, A., Stricker, D.: Learning local patch orientation with a cascade of sparse regressors. In: BMVC, pp. 1–11 (2009)
47. Park, Y., Lepetit, V., Woo, W.: Texture-less object tracking with online training using an rgb-d camera. In: Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on, pp. 121–126. IEEE (2011)
48. ROS: openni_launch_tutorials_intrinsiccalibration - ros wiki (2015). URL http://goo.gl/cEYyaG
49. Rosten, E., Drummond, T.: Machine learning for high-speed corner detection. In: Computer Vision–ECCV 2006, pp. 430–443. Springer (2006)
50. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE International Conference on, pp. 2564–2571. IEEE (2011)
51. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: Robotics and Automation (ICRA), 2011 IEEE International Conference on, pp. 1–4. IEEE (2011)
52. Shotton, J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. Pattern Analysis and Machine Intelligence, IEEE Transactions on **30**(7), 1270–1281 (2008)
53. Suzuki, S., et al.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing **30**(1), 32–46 (1985)
54. Taylor, S., Drummond, T.: Multiple target localisation at over 100 fps. In: Proceedings of the British Machine Vision Conference, pp. 1–11. BMVA (2009)
55. Uchiyama, H., Marchand, E.: Toward augmenting everything: Detecting and tracking geometrical features on planar objects. In: Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on, pp. 17–25. IEEE (2011)
56. Woodfill, J.I., Gordon, G., Buck, R.: Tyzx deepsea high speed stereo vision system. In: Com-

puter Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on, pp. 41–41. IEEE (2004)
57. Wu, C., Clipp, B., Li, X., Frahm, J.M., Pollefeys, M.: 3d model matching with viewpoint-invariant patches (vip). In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. IEEE (2008)
58. Yang, M.Y., Cao, Y., Förstner, W., McDonald, J.: Robust wide baseline scene alignment based on 3d viewpoint normalization. In: Advances in Visual Computing, pp. 654–665. Springer (2010)
59. Zeisl, B., Köser, K., Pollefeys, M.: Viewpoint invariant matching via developable surfaces. In: Computer Vision–ECCV 2012. Workshops and Demonstrations, pp. 62–71. Springer (2012)
60. Zeisl, B., Koser, K., Pollefeys, M.: Automatic registration of rgb-d scans via salient directions. In: Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 2808–2815. IEEE (2013)