D^2CO: Fast and Robust Registration of 3D Textureless Objects using the Directional Chamfer Distance

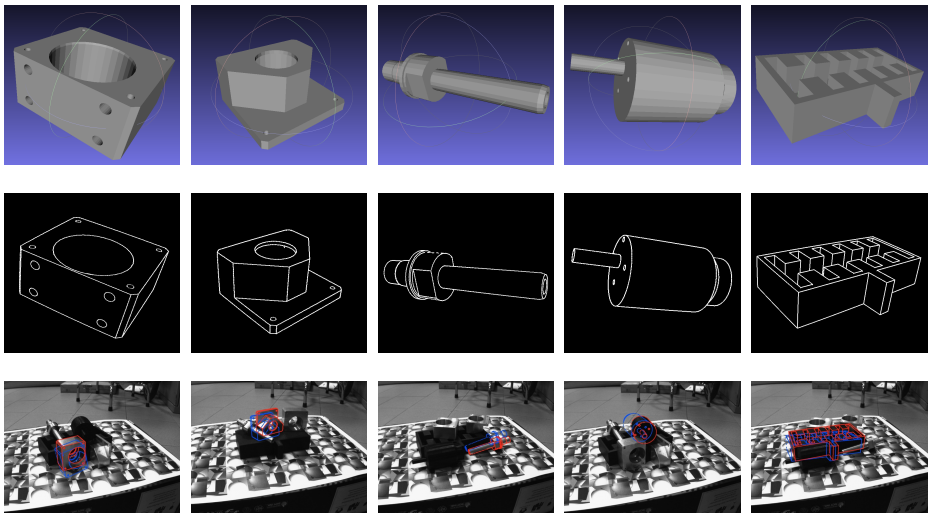
Marco Imperoli and Alberto Pretto

Department of Computer, Control,
and Management Engineering "Antonio Ruberti",
Sapienza University of Rome, Italy
marcoimperoli@gmail.com, pretto@dis.uniroma1.it

Abstract. This paper introduces a robust and efficient vision based method for object detection and 3D pose estimation that exploits a novel edge-based registration algorithm we called Direct Directional Chamfer Optimization (D^2CO). Our approach is able to handle textureless and partially occluded objects and does not require any off-line object learning step. Depth edges and visible patterns extracted from the 3D CAD model of the object are matched against edges detected in the current grey level image by means of a 3D distance transform represented by an image tensor, that encodes the minimum distance to an edge point in a joint direction/location space. D^2CO refines the object position employing a non-linear optimization procedure, where the cost being minimized is extracted directly from the 3D image tensor. Differently from other popular registration algorithms as ICP, that require to constantly update the correspondences between points, our approach does not require any iterative re-association step: the data association is implicitly optimized while inferring the object position. This enables D^2CO to obtain a considerable gain in speed over other registration algorithms while presenting a wider basin of convergence. We tested our system with a set of challenging untextured objects in presence of occlusions and cluttered background, showing accurate results and often outperforming other state-of-the-art methods.

1 Introduction

A reliable object instance detection and localization system is an essential requirement for most robotics applications, from robot-aided manufacturing to service robotics applications, where a robot needs to grasp and manipulate objects in an accurate and reliable way. Many very promising image-based object detection systems usually assumes that the searched objects are characterized by salient visual patterns or textures, to be extracted using robust invariant detectors (e.g., [15]) and matched against precomputed image-based models [18, 23]. Unfortunately, these methods can't handle untextured, non-Lambertian objects: this often prevents the use of these methods for industrial applications,



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

Fig. 1. [Top row]: The 3D CAD models of the untextured objects used in the experiments. (a),(b) and (c) are metal objects with high-reflectance surfaces, while (d) and (e) are black, plastic objects, and present a strong visible-light absorption. [Center row]: The edge templates extracted from the CAD models. [Bottom row]: Some registration results obtained using the D^2CO algorithm, the initial guess is reported in blue, while the final position estimate in red.

where objects are usually untextured and made with non-Lambertian materials as metal or glass. Thanks to the availability of many commercial depth sensor as RGB-D cameras, laser triangulation systems and 3D laser range finders, many vision systems are currently moving toward these 3D sensors (e.g., [20, 21]). Although these systems benefit of a full 3D representation of the workspace, they still have some important limitations: i) The cost of a 3D industrial sensor is still from 3 to 10 times higher than a conventional high resolution industrial camera; ii) Current consumer RGB-D cameras can't handle small objects due to the limited resolution and the minimum viewing distance. Actually, we tested the Microsoft Kinect RGB-D camera with the objects used in our experiments (Fig. 1), and we found that the object (c) is often not even perceived from the depth sensor, even at small distances.

In many cases, edge-based vision algorithm for object detection and localization still provide superior performances. In this context, state-of-the-art methods (e.g., [11, 14]) usually perform an efficient and exhaustive template matching over the whole image. Templates usually represents shapes extracted from the 3D CAD of the object, seen from a number of viewpoints. Unfortunately, in our experience we found that, given as input a single grey level image and no accu-

rate scale priors, none of the tested state-of-the-art matching algorithms provide as first output the best, true-positive, match. Actually, without accurate scale priors, the searching space is huge, imposing a coarse-grained viewpoint discretization. It is usually required to perform many time-consuming object registration steps over a large set of object candidates, extracted from the matching algorithm, in order to accurately detect the true best matches.
To address this problem, in this work we introduce an efficient and robust model-based registration method we called Direct Directional Chamfer Optimization (D^{2}CO). D^{2}CO works on grey level images: the backbone of our method is represented by the 3D distance transform proposed in [14] we call here Directional Chamfer Distance (DCD). This distance improves the accuracy of the Chamfer matching by including edge orientation. The DCD is computed using a 3D image tensor that, for each image pixel coordinates and for each (discretized) direction, provides the minimum distance in a joint direction/location space. In our experience, DCD-based matching provides the best matching results compared with other state-of-the-art matching algorithms in case of clutter and undetected image edges. [14] also introduced a refinement algorithm based on ICP that exploits the DCD to re-compute the point-to-point correspondences. In the following, we will refer to this algorithm with the acronym DC-ICP. DC-ICP provides state-of-the-art registration results, but it is very slow since it requires many iterations to converge. D^{2}CO aims to overcome these limitations, taking advantage of the DCD while boosting the convergence rate. The idea is to refine the parameters using a cost function that exploits the DCD tensor in a *direct way*, i.e., by retrieving the costs and the derivatives directly from the tensor. Being a piecewise smooth function of both the image translation and the (edge) orientation, the DCD ensures a wide basin of convergence. Differently from DC-ICP, D^{2}CO does not require to re-compute the point-to-point correspondences, since the data association is implicitly encoded in the DCD tensor.
Experiments and quantitative evaluations performed on challenging scenes show the effectiveness of our registration method.

1.1 Related Work

Vision systems for learning, detecting and localizing objects have been widely studied in the computer vision and robotics communities for many years.

Image-based systems (e.g., [18, 23]) usually represents objects through collections of distinctive local features (e.g., [15]) extracted form a number of viewpoints. An overview of general image-based object recognition and localization techniques can be found in [22], along with a performance evaluation of many types of visual local descriptors used for 6 DoF pose estimation. Even if this class of methods achieved impressive object recognition and categorization results, they can't be easily applied to textureless objects. Moreover, image-based systems usually fails to obtain accurate object localization results, due to the

discrete nature of the collected viewpoints.

Model-based systems use instead 3D CAD models of the searched objects, or the shapes extracted from the model, seen from a number of viewpoints. This class of methods are well suited for robot-aided manufacturing or service robotics applications: actually, besides ensuring greater accuracy in the localization task, they can deal with untextured objects. In this context, many object detection and registration techniques have been presented.
Recently, we presented a model-based vision system for 3D localization of planar textureless objects [17]. This system exploits a modified Generalized Hough Transform to select object candidates, and an iterative optimize-and-score registration procedure that employs a constrained optimization to be robust against local minima. Although the registration is very accurate, this system can deal only with planar objects rotated up to ± 40 *degrees* around x and y axes.
The iterative closest point (ICP) [2] is probably the best known point registration method: at each iteration, given the current parameters (i.e., a rigid-body transformation), ICP re-computes the correspondences between points and then updates the parameters as a solution of a least square problem. Fitzgibbon [8] proposed to use the Levenberg-Marquardt algorithm to solve the ICP inner loop, while employing a fast distance lookup based on the Chamfer Distance Transform. Jian *et al.* [12] proposed a generalization of the ICP algorithm that represents the input point sets using Gaussian mixture models.
Unfortunately, an initial data association far from the correct one, dramatically increase the chances of ICP to get stuck in a wrong local minima. To mitigate this problem, Gold *et al.* [9] proposed to relax the ICP assumption of fixed correspondences between point sets: they proposed to assign "soft" correspondences between points by means of scalar values in the range [0..1]. This method, called robust point matching (RPM), has been recently extended by Lian and Zhang [13]: they proposed a concave optimization approach for the RPM energy function, that is globally optimal and it can handle the presence of outliers. Unfortunately, these methods include in the parameters set the whole point correspondences matrix, making them often unsuitable to real-world problems.
The Chamfer Distance Transform ([1, 3]), used also for registration in [8], has played an important role in many model-based detection and matching algorithms. Even if the original formulation suffers from not being robust to outliers, Chamfer matching and especially its variations still remain powerful tools used for edge-based object detection and matching. Choi and Christensen [5] employed Chamfer matching inside a particle filtering framework for textureless object detection and tracking. In [19] Shotton *et al.* presented a matching scheme called Oriented Chamfer Matching (OCM): they proposed to augment the Chamfer distance with an additional channel that encodes the edge points orientations. Cai *et al.* [4] used sparse edge-based image descriptor to efficiently prune object-pose hypotheses, and OCM for hypotheses verification. Danielsson *et al.* [7] proposed an object category detection algorithm based on a voting scheme that employs a set of distance transform maps, each one computed for

a discretized orientation. Recently, Liu *et al.* [14] extended this idea proposing the Fast Directional Chamfer Matching (FDCM) scheme, that exploits a 3D distance transform reporting the minimum distance to an edge point in a joint direction/location space: reported results show that FDCM outperform OCM. [14] also presented an ICP-based registration algorithm that exploits the proposed distance transform. The method presented in this paper was inspired from [14] and [8].
Other recent model-based object matching algorithms use spread image gradient orientations saved in a cache memory-friendly way [11] and multi-path edgelet constellations [6].

2 Object Detection

In this section, we describe our object detection approach, that exploits the DCD tensor in order to extract a set of objects candidates (i.e., their 'rough' 3D locations) from the input image, based on the given model template.

2.1 Object Model

Edges represent the most informative image features that characterize untextured objects: edges are usually generated by occlusions (depth edges) and high curvature. Given a 3D CAD model of an object (e.g., first row of Fig. 1), we need to extract a 3D template that includes only these edges. We start from the 3D model wireframe, preserving only edges that belong to high curvature parts or to the external object shape, while using the OpenGL z-buffer to deal with occlusions. Some results of this procedure are shown in the second row of Fig. 1. It is important to note that, in the general case, this procedure should be repeated for each viewpoint, i.e., for each object position.
We finally produce a *rasterization* of this template, i.e. we extract from the template a set of m sample points $\mathcal{M} = \{\mathbf{o}_1, \ldots, \mathbf{o}_m\} \in \mathbb{R}^3$, in the object reference frame. Typically we employ a rasterization step of $1 - 2\ mm$. We also collect another set of m points, $\bar{\mathcal{M}} = \{\bar{\mathbf{o}}_1, \ldots, \bar{\mathbf{o}}_m\} \in \mathbb{R}^3$, where:

$$\bar{\mathbf{o}}_i = \mathbf{o}_i + \hat{\tau}(\mathbf{o}_i) \cdot dr \tag{1}$$

$\hat{\tau}(\mathbf{o}_i)$ is a function that provides the unit tangent vector, i.e. the unit direction vector of the 3D edge the raster point belongs to, while dr is a small scalar increment, $dr \ll 1$. Given a transformation $\mathbf{g}_{cam,obj} \in \mathbb{SE}(3)$ from the object frame to the camera frame, we can project the raster points on the image plane as:

$$\mathbf{x}_i = \pi(\mathbf{o}_i, \mathbf{g}_{cam,obj}) \in \mathbb{R}^2, i = \{1, \ldots, m\} \tag{2}$$

The idea behind these two set of 3D points is simple: by projecting on the image plane the raster points $\mathcal{M}$ along with the points $\bar{\mathcal{M}}$, it is possible to easily recover also the 2D local directions (orientations) of the projected edge points in the image plane.

2.2 Edge Points Extraction

In order to match the edge template extracted from the CAD model, we need to detect edges in the input image. We adopt here the concept of *edgelet*, a straight segment that can be part of a longer, possibly curved, line, extracted using a state-of-the-art detection algorithm for line segment detector called LSD [10]. This algorithm searches the input image for edgelets starting from pixels with higher gradient magnitude, looking in the neighbourhood for pixels with similar gradient directions. A line-support region is therefore selected and validated in order to discard the outliers.
We employ the LSD algorithm on a Gaussian pyramid of the input image, enabling to include in the final set also edgelets that appear in higher scales. This technique increases the sensitivity of the edge detector, at the cost of a reduced accuracy in the localization of the edgelets. For each detected edgelet, we also compute its orientation in the image reference frame. We define as $\mathcal{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\} \in \mathbb{R}^2$ the set of pixels (*edgels*) that belong to edgelets.

2.3 Directional Chamfer Distance

As introduced above, our approach leverages the Directional Chamfer Distance tensor in both the detection and registration steps. The DCD tensor ($DT3_V$) is represented by an ordered sequence of distance transform maps[1], each one representing a *discretized edge direction*. The basic idea behind the Directional Chamfer Distance is simple:

- Divide the set of edgelets computed in Sec. 2.2 into several subsets by quantizing their directions;
- draw each edgelets set in a different binary image;
- compute one distance transform map for each subset using the binary images computed above.

In this way, each map reports the minimum distance from a set of edges with almost the same direction. Liu *et al.* [14] extended this idea enabling the DCD tensor $DT3_V$ to encode the minimum distance to an edge point in a joint location and orientation space. Be $\mathbf{x}_i$ the 2D projection on the image plane of a 3D raster edge point $\mathbf{o}_i$ with $\Xi(\mathbf{x}_i)$ a function that provides its (scalar) direction in the image reference frame, computed projecting also $\bar{\mathbf{o}}_i$ (see Sec. 3). The minimum distance to an edge point (edgel) $\mathbf{e}_j$ can be recovered as:

$$DT3_V(\mathbf{x}_i, \Xi(\mathbf{x}_i)) = \min_{\mathbf{e}_j \in \mathcal{E}} (\|\mathbf{x}_i - \mathbf{e}_j\| + \lambda \|\phi(\Xi(\mathbf{x}_i)) - \phi(o(\mathbf{e}_j))\|) \tag{3}$$

where $o(\mathbf{e}_j)$ is the edgel orientation and $\phi(.)$ is an operator that provides the nearest quantized orientation. The tensor $DT3_V$ can be easily computed by applying a forward and backward recursions to the sequence of distance transform

[1] In our case, a distance transform is an image where each pixel reports the distance to the closest edge pixel (edgel).

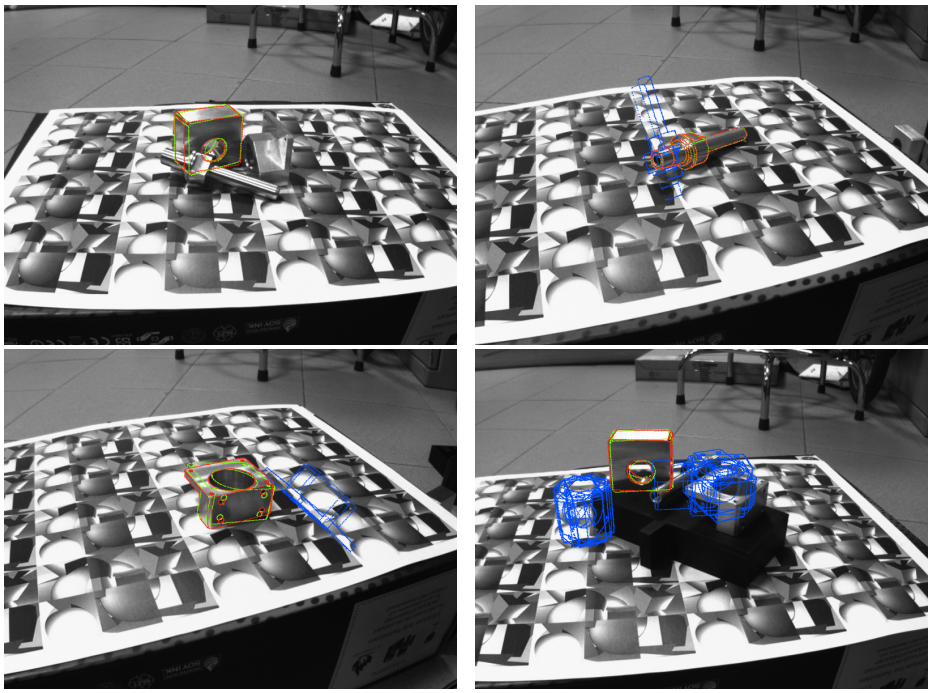


Fig. 2. Some examples of objects candidates extraction. The red template represents the ground truth position of the object, green and blue templates represent the true and false positives, respectively.

maps described above, see [14] for the details. Since our optimization framework employs the tensor $DT3_V$ in a direct way (see Sec. 3), we need to ensure that the function $DT3_V : \mathbb{R}^3 \rightarrow \mathbb{R}$ that it represents is piecewise smooth. To this end, we smooth the tensor along the direction (orientation) dimension using a simple Gaussian filter. In all our experiments, we use 60 discretized orientations and we set λ to 100 and σ^2 (the variance the Gaussian filter) to 1.

2.4 Candidate Extraction

Since we perform the object detection task without knowing any accurate scale prior, the huge 6D searching space imposes a coarse-grained viewpoint discretization. In order to speedup the process, for each object in the dataset we precompute the (projected) raster templates along with their image orientations for a large number of possible 3D locations. Each template includes in such a way a set of image points along with their orientation: by performing a set of lookups on the tensor $DT3_V$, we can compute the average distance in an efficient way. Finally, we sort the templates for increasing distance: the top rated tem-

plates (e.g., see Fig. 2) represent our objects hypothesis (or "object candidates"), to be registered and validated.

3 Object Registration

Once we have obtained a set of objects candidates, we need to precisely locate each true positive object, while discarding the outliers. D^{2}CO refines the object position employing a non-linear optimization procedure that minimizes a tensor-based cost function.

Given a set of m raster points extracted from the 3D CAD model, from Eq. 2 we can obtain the corresponding image projections $\mathbf{x}_i$. We can also express the transformation $\mathbf{g}_{cam,obj}$ in terms of a translation vector $\mathbf{T} = [t_x \ t_y \ t_z]^T$ and a orientation vector $\mathbf{\Omega} = [r_x \ r_y \ r_z]^T$, both in $\mathbb{R}^3$. We make explicit this fact using the notation $\mathbf{g}(\mathbf{T}, \mathbf{\Omega}) = \mathbf{g}_{cam,obj}$. $\mathbf{R}(\mathbf{\Omega}) \doteq \exp(\widehat{\mathbf{\Omega}})$ is the rotation matrix corresponding to the rotation vector $\mathbf{\Omega}$, where $\widehat{\mathbf{\Omega}}$ is the skew-symmetric matrix corresponding to $\mathbf{\Omega}$, and $Log_{\mathbb{SO}(3)}(\mathbf{R}(\mathbf{\Omega})) \doteq \mathbf{\Omega}$ is the rotation vector $\mathbf{\Omega}$ corresponding to the rotation matrix $\mathbf{R}(\mathbf{\Omega})$ [16]. Our optimization procedure aims to find the parameters $(\tilde{\mathbf{T}}, \tilde{\mathbf{\Omega}}) \in \mathbb{R}^6$ that minimize:

$$E(\mathbf{T}, \mathbf{\Omega}) = \frac{1}{2} \sum_{i=1}^{m} DT3_V \left[\pi(\mathbf{o}_i, \mathbf{g}(\mathbf{T}, \mathbf{\Omega})), \Xi(\pi(\mathbf{o}_i, \mathbf{g}(\mathbf{T}, \mathbf{\Omega})))\right]^2 \tag{4}$$

While we can assume that, for small viewpoint transformations, the 3D raster points do not change (i.e., we can neglect changes in the occlusions), this fact does not apply for their image projections. Moreover, Eq. 4 also requires to constantly update the projected (edge) point orientations. We first project also the point set $\bar{\mathcal{M}}$: $\bar{\mathbf{x}}_i = \pi(\bar{\mathbf{o}}_i, \mathbf{g}_{cam,obj}), i = \{1, \ldots, m\}$. If we define $\mathbf{d}_i \triangleq \bar{\mathbf{x}}_i - \mathbf{x}_i$, for each iteration of the optimization we can compute the updated orientations as:

$$\Xi(\pi(\mathbf{o}_i, \mathbf{g}(\mathbf{T}, \mathbf{\Omega}))) = \Xi(\mathbf{x}_i) = atan\left(\frac{\mathbf{d}_i(1)}{\mathbf{d}_i(0)}\right) \tag{5}$$

In order to apply a non-linear minimization on $E(\mathbf{T}, \mathbf{\Omega})$, we have to compute its derivatives ∇E. Application of the chain rule yields:

$$\nabla E = \sum_{i=1}^{m} \{\nabla DT3_V \ \nabla \left[\pi(\mathbf{o}_i, \mathbf{g}(\mathbf{T}, \mathbf{\Omega})), \Xi(\pi(\mathbf{o}_i, \mathbf{g}(\mathbf{T}, \mathbf{\Omega})))\right]\} \tag{6}$$

Since $DT3_V$ is only defined at discrete points, its derivatives $\nabla DT3_V$ should be computed in a numerical, approximate way. To this end, we compute the x and y derivatives as the image derivatives of the currently selected distance map and, in a similar way, the derivative along the orientation direction ξ as:

$$\frac{\delta DT3_V}{\delta o}(\mathbf{x}, \xi) = \frac{DT3_V(\mathbf{x}, \xi + 1) - DT3_V(\mathbf{x}, \xi - 1)}{2} \tag{7}$$

We lookup the $DT3_V$ tensor employing a bilinear interpolation operator, adapted to the 3D nature of $DT3_V$: this enables to improve the level of smoothness of

the cost function.
We perform the optimization using the Levenberg-Marquardt algorithm and, as suggested in [8], a Huber loss function in order to reduces the influence of outliers.

3.1 The Scoring Function

Some of the selected hypothesis used as initial guess for the registration may represent false positive objects: after the position refinement presented above, we need to employ a metrics that allows us to discard the outliers and to select the best match. We use a simple but effective scoring function based on local image gradient directions. Given $\mathbf{x}_i$ a raster point projected on the image plane, we can compute its image normal direction $n_{dir}(\mathbf{x}_i)$: in the ideal case of a perfect match, this direction should correspond to the local gradient direction $\mathbf{I}_\theta(\mathbf{x}_i)$ (up to a rotation of π *radians*), where $\mathbf{I}_\theta$ is the gradient direction image computed directly from the input image. We define the scoring function as:

$$\Psi(\mathbf{g}_{cam,obj}) = \frac{1}{m}\sum_{1}^{m} |\cos\left(\mathbf{I}_\theta(\mathbf{x}_i) - n_{dir}(\mathbf{x}_i)\right)| \tag{8}$$

Clearly $\Psi(\mathbf{g}_{cam,obj})$ can get values from 0 to 1, where 1 represents the score for a perfect, ideal match. In our experiments, all good matches (inliers) obtain a score greater than 0.8. We discard all the matches with a score less than this threshold: they can be outliers (false positive objects) or real objects heavily occluded.

4 Experiments

We present two different experimental validations. The first experiment aims to show that FDCM [14], that is the backbone of D^{2}CO, outperforms in our dataset another recent state-of-the-art object detection algorithm. In a second experiments, we compare D^{2}CO to other registration techniques, showing state-of-the-art performances with a gain in speed up to a factor 10.

Our dataset[2] is composed by 60 1024x768 grey level images of scenes that contain up to 5 untextured objects[3] (see Fig. 1) disposed in arbitrary 3D positions, often mutually occluded. In some images we added a background board with multiple patterns in order to simulate a (visual) crowded background (e.g., see Fig. 4). Each image is provided with the ground truth positions of the objects, obtained with an externally guided procedure. All the experiments were performed running the algorithms on a standard laptop with an Intel Core i7-3820QM CPU with 2.70GHz, using a single core. All the compared algorithms

[2] The dataset is available at `http://www.dis.uniroma1.it/~labrococo/D2CO`
[3] The objects used in the experiments are currently employed in the RoCKIn@Work competitions, `http://rockinrobotchallenge.eu`

has been implemented in C++ without any strong optimization. When possible, they share the same codebase and the same parameters, enabling an objective performance and timing comparisons.

4.1 Object Detection

In the first experiment, we compare our object matching approach, that is strongly related with FDCM presented in [14], with a state-of-the-art object detection approach (called in the plots GRM, from Gradient Response Maps) described in [11] (since we are using only images, we use the LINE-2D version). In our experiments we don't perform any memory linearization as in [11], since we don't assume that the object projection remain the same every $x - y$ translation. Anyhow, this modification does not affect the performance of GRM: it just runs slower.
We perform object matching starting from a set of 1260 pose candidates for each one of the 60 scenes of our dataset. Each of these candidates are acquired by sampling a large cubic scene region containing the target objects. Fig. 3 shows how FDCM outperforms GRM in terms of correct detection rate against the number of false positives for each image.

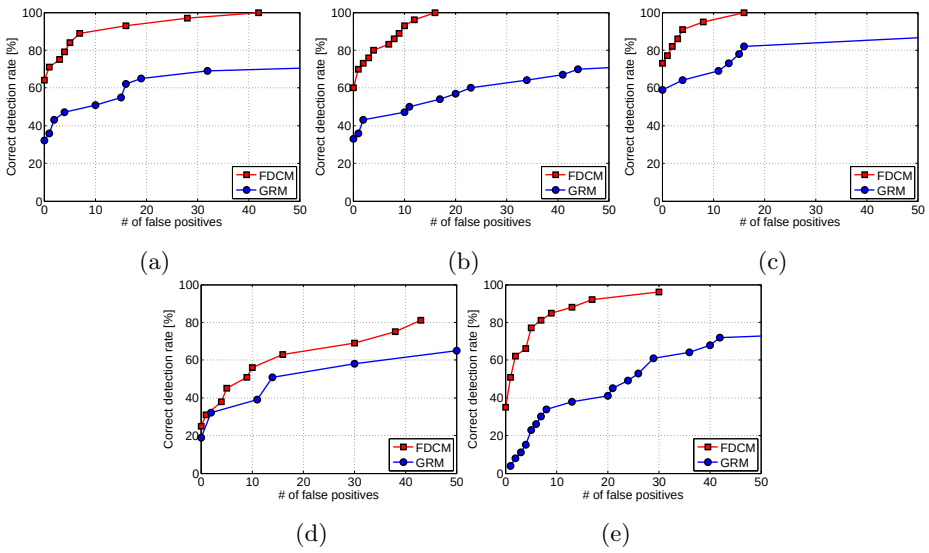
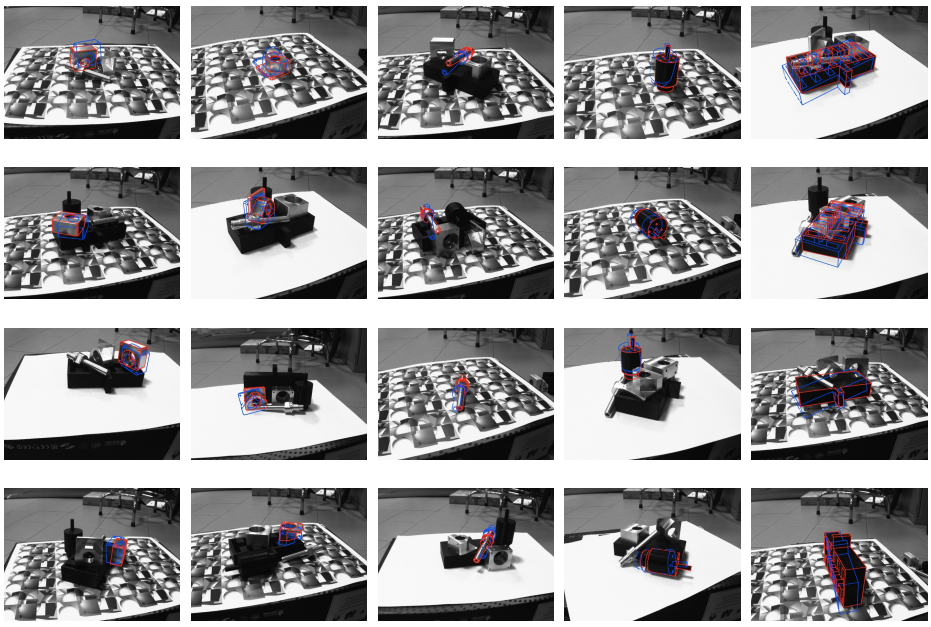


Fig. 3. True positives rate plotted against the number of false positives (the letters (a),..(e) refer to the objects of Fig. 1).



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

Fig. 4. Other registration results obtained using the D^{2}CO algorithm, the initial guess is reported in blue, final position estimate in red (the letters (a),..(e) refer to the objects of Fig. 1).

4.2 Object Registration

We compared our approach (D^{2}CO) to DC-ICP [14], LM-ICP [8], C-ICP (ICP that exploits the Chamfer distance) and Direct (a simple coarse-to-fine object registration strategy that uses a Gaussian pyramid of gradient magnitudes images). All the tested algorithms share the same inner loop's stopping criteria parameters. We set the number of external ICP iterations to 50: we verified that this is a good trade-off in order to reach reliable results in our dataset. For each image and for each object in the image, we sampled 100 random positions around the ground truth position (e.g., the blue templates reported in Fig. 4). We used these positions as initial guesses for the registration algorithms we are testing. The final estimated position (e.g., the red templates reported in Fig. 4) is checked against the ground truth: if the total angular error was less than 0.1 radians, and the total error of translation was less than 5 mm, the registration was considered correct.
The proposed D^{2}CO algorithm outperforms the other methods in almost all tests (Fig. 5), while getting a gain in speed of a factor 10 compared to the second most competitive approach (see Table 1).

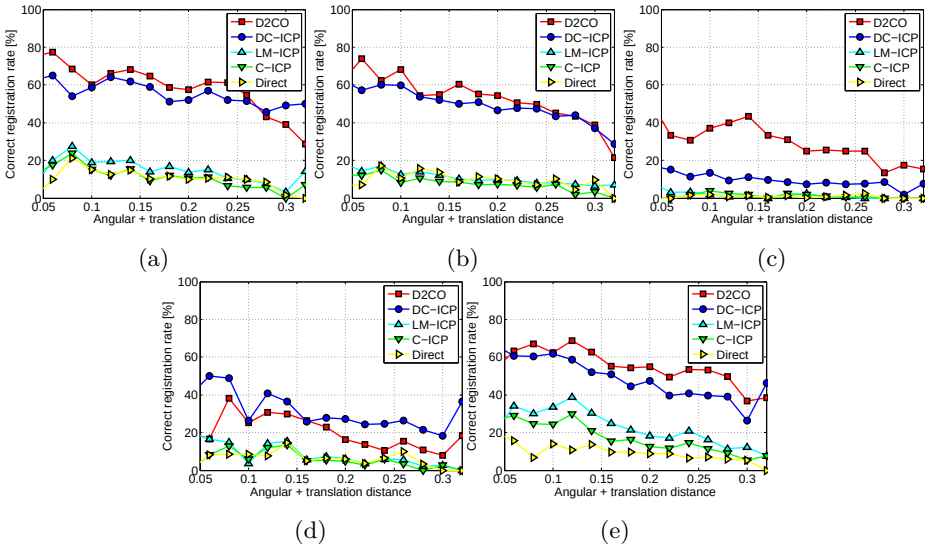


Fig. 5. Correct registrations rate plotted against the distance (angle + translation) of the initial guess from the ground truth position (the letters (a),..(e) refer to the objects of Fig. 1).

Table 1. Average object registration time (milliseconds).

Algorithm	**D^{2}CO**	**DC-ICP**	**LM-ICP**	**C-ICP**	**Direct**
Time (msec)	56.49	659.09	68.43	601.10	65.83

5 Conclusions and Future Works

In this paper, we have proposed an object detection and localization system that combines a state-of-the-art untextured object detection algorithm with a novel registration strategy that leverages the Directional Chamfer Distance tensor in a direct and efficient way. We tested our system with a set of challenging untextured objects in presence of occlusions and cluttered background, getting state-of-the-art results while saving computation time.
We are currently improving our system by enabling the optimization to work directly on the directional integral images representation of the DCD tensor, in order to further speedup the registration process.

Acknowledgement. This research has been supported by the European Commission under: 644227-Flourish and FP7-ICT-601012 (RoCKIn Project).

References

1. Harry G. Barrow, Jay M. Tenenbaum, Robert C. Bolles, and Helen C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. of 5th International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.
2. Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.
3. Gunilla Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):849–865, 1988.
4. Hongping Cai, Tomás Werner, and Jiri Matas. Fast detection of multiple textureless 3-d objects. In *Proc. of: 9th International Conference Computer Vision Systems*, pages 103–112, 2013.
5. Changhyun Choi and Henrik I. Christensen. 3d textureless object detection and tracking: An edge-based approach. In *Proc. of: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3877–3884, 2012.
6. Dima Damen, Pished Bunnun, Andrew Calway, and Walterio W. Mayol-Cuevas. Real-time learning and detection of 3d texture-less objects: A scalable approach. In *Proc. of: British Machine Vision Conference*, pages 1–12, 2012.
7. Oscar M. Danielsson, Stefan Carlsson, and Josephine Sullivan. Automatic learning and extraction of multi-local features. In *Proc. of IEEE 12th International Conference on Computer Vision*, pages 917–924, 2009.
8. A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. In *British Machine Vision Conference*, pages 662–670, 2001.
9. Steven Gold, Anand Rangarajan, Chien ping Lu, and Eric Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern Recognition*, 31:957–964, 1997.
10. Rafael Grompone von Gioi, Jeremie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(4):722–732, 2010.
11. Stefan Hinterstoisser, Cedric Cagniart, Slobodan Ilic, Peter F. Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):876–888, 2012.
12. Bing Jian and Baba C. Vemuri. Robust point set registration using gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1633–1645, 2011.
13. Wei Lian and Lei Zhang. Point matching in the presence of outliers in both point sets: A concave optimization approach. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 352–359, 2014.
14. Ming-Yu Liu, Oncel Tuzel, Ashok Veeraraghavan, Yuichi Taguchi, Tim K. Marks, and Rama Chellappa. Fast object localization and pose estimation in heavy clutter for robotic bin picking. *I. J. Robotic Res.*, 31(8):951–973, 2012.
15. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
16. Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An invitation to 3D vision, from images to models*. Springer Verlag, 2003.
17. A. Pretto, S. Tonello, and E. Menegatti. Flexible 3d localization of planar objects for industrial bin-picking with monocamera vision system. In *Proc. of: IEEE International Conference on Automation Science and Engineering (CASE)*, pages 168 – 175, 2013.

18. Silvio Savarese and Fei-Fei Li. 3d generic object categorization, localization and pose estimation. In *Proc. of IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.
19. Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1270–1281, 2008.
20. Min Sun, Shyam Sunder Kumar, Gary R. Bradski, and Silvio Savarese. Object detection, shape recovery, and 3d modelling by depth-encoded hough voting. *Computer Vision and Image Understanding*, 117(9):1190–1202, 2013.
21. Jie Tang, Stephen Miller, Arjun Singh, and Pieter Abbeel. A textured object recognition pipeline for color and depth image data. In *Proc. of the IEEE International Conference on Robotics and Automation*, 2012.
22. Fredrik Viksten, Per-Erik Forssén, Björn Johansson, and Anders Moe. Comparison of local image descriptors for full 6 degree-of-freedom pose estimation. In *Proc. of the IEEE international conference on Robotics and Automation*, 2009.
23. Tom Yeh, John J. Lee, and Trevor Darrell. Fast concurrent object localization and recognition. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 280 – 287, 2009.