Multi-View Object Instance Recognition in an Industrial Context

Mustafa, Wail; Pugeault, Nicolas ; Buch, Anders Glent; Krüger, Norbert

Published in:
Robotica

DOI:
10.1017/S0263574715000430

Publication date:
2017

Document version
Submitted manuscript

Document license
Unspecified

Citation for pulished version (APA):
Mustafa, W., Pugeault, N., Buch, A. G., & Krüger, N. (2017). Multi-View Object Instance Recognition in an Industrial Context. Robotica, 35(2), 271-292 . DOI: 10.1017/S0263574715000430

Multi-View Object Instance Recognition in an Industrial Context

Wail Mustafa*, Nicolas Pugeault†, Anders G. Buch* and Norbert Krüger*
*Mærsk Mc-Kinney Møller Institute University of Southern Denmark,
Campusvej 55, 5000 Odense C, Denmark
Email: wail@mmmi.sdu.dk
†Centre for Vision, Speech and Signal Processing, Faculty of Engineering & Physical Sciences, University of Surrey,
Guildford GU2 7XH, United Kingdom

***Abstract*—We present a fast object recognition system coding shape by viewpoint invariant geometric relations and appearance information. In our advanced industrial work-cell, the system can observe the work space of the robot by three pairs of Kinect and stereo cameras allowing for reliable and complete object information. From these sensors, we derive global viewpoint invariant shape features and robust color features making use of color normalization techniques.**

We show that in such a set-up, our system can achieve high performance already with a very low number of training samples, which is crucial for user acceptance and that the use of multiple views is crucial for performance. This indicates that our approach can be used in controlled but realistic industrial contexts that require—besides high reliability—fast processing and an intuitive and easy use at the end-user side.

I. Introduction

The task of object recognition in an industrial assembly set-up (as shown e.g., in Fig. 1) is fundamentally different from the 'general object recognition problem' from 2D images as addressed, for instance, in the Pascal Challenge [1]. It also differs largely from object recognition problems posed by 3D datasets such as [2], which have been in particular recently discussed with the availability of cheap RGBD sensors, such as the Kinect camera[1]. The main difference for an industrial set-up is that the sensors and the number thereof can be chosen freely as well as the fact that illumination can be controlled to a large degree by the light sources on the platform or alternatively, that color can be calibrated by color normalization techniques. A particular challenge is that the goal of performing large sequences of actions in assembly processes requires very reliable and also fast object recognition and localization as well as intuitive use at the end-user side.

In this paper, we address the task of object recognition in a well-controlled scenario assuming objects occurring only in the rather restricted work space of the robot as shown in Fig. 1b. The set-up resembles an 'intelligent work-cell' in an advanced production scenario. The task at hand is to determine the presence of objects in the working space covered by three pairs of Kinect and stereo cameras. In contrast to the object recognition problem addressed by standard databases operating on 2D images or 3D depth information extracted from individual views, in our set-up we can operate on rather complete 3D data computed by three different views arranged in a triangle (see Fig. 1a). Our method is then supposed to be used to trigger other mechanisms such as pose estimation or manipulation actions (e.g., grasping, peg-in-hole, or screwing actions) as well as monitoring such processes in the context of complex assembly operations (see, e.g., [3]).

In this paper, 3D *texlets* (see Sect. III-D) serve as basic visual representations of objects. These texlets are acquired by two different sensors—stereo and Kinect cameras—simultaneously. From these, viewpoint invariant representations based on appearance and geometric relations are computed. The use of 3D information is attractive since it allows us to extract viewpoint invariant features in terms of geometric relations (such as distance or angle) between 3D entities. The fact that we operate in a limited and controlled workspace leads to reliable 3D shape and appearance information. In our representations, both aspects—shape and color—are represented separately, allowing us to investigate their relative importance. This space of feature relations (in the following also called 'relational space') can be expressed in (potentially high-dimensional) histograms providing unique and interpretable descriptors for specific objects (e.g., the distance between two parallel surfaces, see Subsect. III) which, besides being viewpoint invariant, is also rather specific for a certain object. As we will show in this paper, this is useful for efficient learning because a relatively few object recordings are required to learn representations for reliable object recognition.

As a classification algorithm, multi-class Random Forest [4], [5] is applied in this paper. Random Forests (RFs) have been found to be efficient because they combine the simplicity of decision trees with the stability of voting methods The algorithm is trained with a set of real objects represented by a combination of their relations and appearance histograms (see Sect. III).

The main achievements of our work can be summarized as follows:

- we demonstrate the potential of applying 3D viewpoint invariant relations by achieving high-performance classification with very few training samples. The remaining misclassifications are caused mainly by object pairs

[1] http://www.xbox.com/kinect

with very fine shape and color differences. For these objects, the sensor resolution simply does not allow for the required precision for coding the shape and color differences.

- we can achieve a significant improvement in performance by using multiple cameras comparing to single—or even two—cameras. This is due to the fact that significant aspects of objects are expressed in our representation by relations, which only manifest themselves with a rather complete 3D representation only achievable by means of three views from different perspectives.
- we show that our approach, when applied to Kinect sensor data, has a much better performance in comparison with the sensor data extracted by standard stereo cameras.
- we show that, even under varying illumination conditions, it is possible to derive strong appearance features from color information when a color normalization step prior to the classification is performed.
- we show that the combination of color and shape information leads to higher recognition results, hence both features are complementary.
- we show how our approach can be used as a trigger for pose estimation and by that complex scene description in terms of object identity and object pose can be computed.

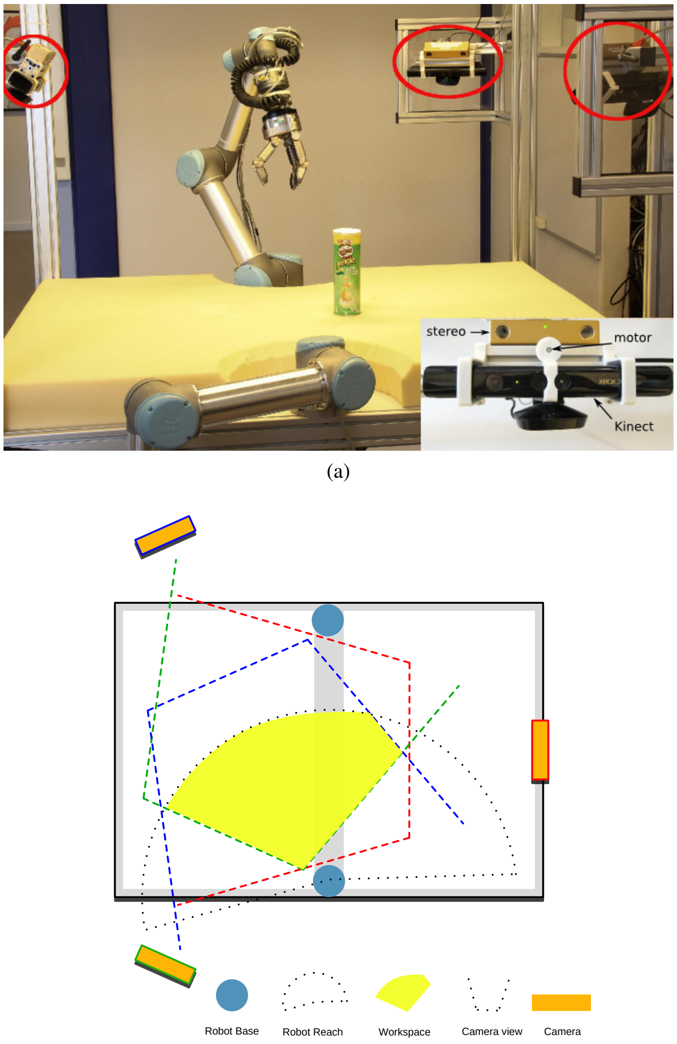
This work is based on a representation introduced in a conference paper [6]. In this journal paper, we however go significantly beyond the work in [6] in multiple respects. First, we apply our approach to a larger and significantly more difficult object set [2]. Second, we investigate the representation in terms of two crucial parameters connected to binning and smoothing. Third, we investigate the effect of color normalization. Fourth, instead of using only 1D and 2D histograms, we also make use of higher dimensional representations and finally we combine our representation with a pose estimation step allowing for a complete description of complex scenes in terms of object identity and pose.

This paper is organized as follows: Sect. II discusses the state of the art. Sect. III describes in details the object recognition system introduced in this paper. In Sect. IV, we present a benchmark dataset, describe the experiments performed on the system and show the results. Sect. V presents an application scenario in which the object recognition system is used to trigger a pose estimation task and by that allow for the interpretation of complex scenes. Finally, a conclusion is given in Sect. VI

II. State of the Art

We first discuss the state of the art of the general problem of object recognition and then we focus on this problem in an industrial context.

Object recognition and classification learning: The problem of object recognition and classification has been intensively studied over the last decades. The annual Pascal challenges (see, e.g., [1], [7]) promote rigorous evaluation and comparison of object recognition algorithms. Although significant successes have been reported, criticism has been raised that the typical visual class recognition may learn pose and context-specific features rather than the object itself. In that context, notably Nicolas Pinto and colleagues showed that a simple model of the V1 cortical area of the human brain could perform well on a typical natural image benchmark [8]. Also, the generalization of classifiers or detectors learned on a specific dataset to another, called domain adaptation, remains a challenge [9]. In contrast to those works, this article is not concerned with the detection and recognition of objects ”in the wild”, but rather with the reliable and fast recognition of objects in a specific industrial set-up, for which we however cannot assume consistency in context. Also the number of training examples is supposed to be kept very low since the requirement of recording a large training set would increase



(a)

(b)

Fig. 1. The set-up. (a) overview of the set-up showing the robot arms and the camera pairs. The close-up view shows one pair of stereo and Kinect (with a vibration-inducing motor attached to it). (b) a top-view sketch for the set-up depicting the workspace.

[2] This data set is available at http://caro.sdu.dk/sdu-dataset (we will make it available by the final submission of this manuscript)

the complexity of the application of such a system at the end-user side.

Classification is typically done in two steps, feature extraction and classification, where the first step extracts or learns a set of features or parts to describe the objects' training samples and a second step which associate an object class to a new unseen object sample. The features used typically describe local image patches, often chosen for robustness to affine transformation, e.g., SIFT [10]. Alternatively, feature descriptors based on relative shape information, called 'shape context' were proposed by Belongie et al. [11]. The shape context of a point encodes the relative distribution of other points on the shape. It has been used as such to perform point-to-point matching in 2D. In [12], the shape context was extended to 3D and defined for a local neighborhood.

After feature extraction, classification can be done in two ways: the first class of methods effectively performs image *retrieval* and is based on nearest-neighbor matching (e.g., [10]); the second makes use of discriminative classification algorithms (such as Support Vector Machines [13] or Boosting [14]). Generally, discriminative approaches lead to higher classification performance, but can suffer from poor generalization when using weak visual features or when the variety of the training data is too limited.

Recently, hierarchical approaches such as convolutional networks have shown high performance (at the price of a significant computational cost) on such large dataset as ImageNet [15]. Interestingly, it was shown that the hierarchies learned on this dataset could then perform well when applied on a different dataset [16], offering some hope for solving the domain adaptation problem. It is worth noting that these results are based on very large training data and obtained at a significant computational cost. Both the computation time and the necessity to create large training data cause significant hurdles for the application of such systems in an industrial context.

The approach used in this work differs in particular in two aspects from the approaches to object recognition discussed above: First, the system is based on a multi-view set-up that is specific to an industrial scenario, aiming at high level recognition performance; Second, this set-up allows us to develop a feature describing the objects' 3D-shape in a pose-invariant fashion allowing the robust use of discriminative classification methods. As a consequence, a small amount of training data is required to achieve good performance.

Object recognition in industrial setup: Object recognition has been used in industrial production set-ups mainly for the identification of a small set of objects (mostly less than five objects) and is in general used as a prior step for pose estimation. Such approaches are nowadays part of standard vision softwares such as Cognex[3], Scorpion Vision[4] and Matrox[5]. These systems mostly provide 2D approaches. This necessarily leads to a larger complexity in using these systems, since the projective map need to be accounted for in the set-up of cameras. This requires covering all possible viewpoint and appearance changes by the training set as well as handling quite a number of parameters in the software that need to be adapted. Recently, also approaches using 3D data have evolved [17], but such approaches have not, to our knowledge, been used on industrial vision systems for object recognition tasks.

Although vision gradually enters production units, statements from end-users and even robot integrators such as ”vision does not work” are not uncommon. Such statements are usually caused by the fact that the use of the applied vision software requires at least some expert knowledge about the involved visual processes and the camera geometry. As argued above, the use of 3D vision approaches—as done in our work—can facilitate the application of vision algorithms in industrial scenarios by reducing the complexity introduced by viewpoint changes caused by the projective map, or in other words, by allowing the end-user to operate in the more intuitive Euclidean space.

Another advantage in an industrial context compared to the general object recognition problem discussed above is that the actual camera set-up can be freely chosen. This opens the possibility to increase robustness by using multiple cameras. In addition, due to relatively short distances between camera and object, 3D sensors such as Kinect like cameras can be used. The novelty of our approach lies in the explicit use of multiple simultaneously recorded views, utilizing viewpoint invariant relations that can only be generated based on the combination of all three views.

Another aspect of our approach is that due to the pose invariant representation only few training examples are required to achieve a high recognition performance. This facilitates the often quite sophisticated training that is in general required for view based systems (see, e.g., [2]).

In section IV-D, we will show that we can achieve with very few training samples high object recognition performance in a controlled—but, from a point of view of industrial production, realistic—environment for a recognition task which is much harder than it usually occurs in an industrial setting. This allows systems to perform object recognition for assembly processes with some complexity in an industrial context based on visual information.

III. Object Instance Recognition System

In this section, we describe in details the components of the object recognition system introduced in this article.

A. System overview

Fig. 2 shows the system components. The system operates on the robot platform described in Subsect. III-B in which three views are captured by three sensors (Kinect or stereo). For a single view, the process starts with applying colorimetric camera calibration on images as explained in Subsect. III-C. This is followed by scene preprocessing for table removal and object segmentation. The table removal is applied using a

[3] http://www.cognex.com

[4] http://scorpionvision.co.uk

[5] http://www.matrox.com

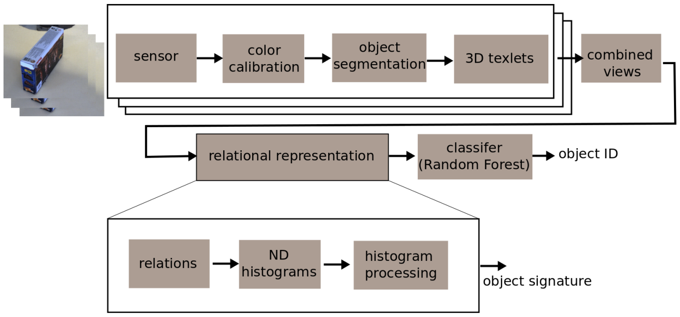


Fig. 2. Object instance recognition: block diagram of the different components. The three layers on top shows the components that process the sensory data from each single camera. The lower components are the ones process the 3D combined data where the relational representation of objects is obtained to form the object signature, which is then passed to the supervised classification algorithm.

RANSAC-based plane detection whereas the segmentation is performed using 3D Euclidean clustering. All this is performed in the 3D point cloud data using PCL library [18].

For a segmented object, the 3D texlets features described in Subsect. III-D are then extracted forming a single 3D view of the object. Using the relative camera transformations, which are estimated through external camera calibration, the three views are combined in the 3D space. These combined features form the 3D representation of objects from which the relational representation is computed.

The relational representation is a pose-invariant object description obtained by computing shape and color relations from pairs of 3D features, see Subsect. III-E for details. The different relations are then binned in multi-dimension (ND) histograms to form the object signature (Subsect. III-F). Optionally, histograms are processed by means of spatial filtering for noise reduction (Subsect. III-G). The resulting object signature is finally passed to a classifier; Random Forest (Subsect. III-H) is used here. During the training phase, which is performed in a supervised manner, a classification model of decision trees is created from the training data. The model is used to predict the object ID (with an associated conference value) during execution (i.e., prediction phase).

B. Multi-view sensors (set-up)

The environment in which we want to solve the object recognition task is a robot work-cell (which can be used e.g., in industrial assembly processes as in [19]). Fig. 1a shows an overview of the set-up and camera pairs in use. The work-cell consists of two robot arms performing manipulation tasks with a variety of objects. Three pairs of Kinect and stereo cameras are mounted in a close to equilateral triangular configuration. By combining the three views of this set-up, we obtain a complete (except for the surface in contact with table) representation of the objects' 3D shape. Note that a vibration-inducing motor is attached to each Kinect to reduce the interference effects occurring when multiple Kinects with overlapping views are simultaneously used [20].

Fig. 1b is a sketch (plan view) of the set-up, showing the field of view of each camera and the area of reach of the main robot arm. The yellow-shaded area depicts the workspace in which our system operates. The workspace is defined by the intersection of the three fields of view and the area of reach. The requirement that all cameras cover the area is strictly limiting the usable workspace. On the other hand, for complicated manipulation tasks, such as the ones this set-up is intended for, high performance of object recognition and pose estimation is needed. In this paper, we show that having multiple views enhances performance significantly by providing a complete 3D representation, which allows for encoding a rich set of relations unavailable from single views (e.g., opposite surfaces). Furthermore, such a multi-view approach also increases the system's robustness against occlusion.

C. Colorimetric camera calibration

One way to increase color robustness is to apply colorimetric camera calibration (see Fig. 3). By doing so, we minimize two effects causing instability of color features. First, the variation in illumination due to having different lighting conditions. The second effect is the variation in the color representation that may occur due to different sensors. On the system level, this process leads to a more robust object instance recognition based on color (see Sect. IV-C).

Essentially, the process involves reading reference color values obtained from the image and do the correction based on their true values. These reference colors are presented in a color checker[6] lying within the sensor's field of view. The color checker contains 24 color patches representing natural and gray-scale colors, which was first introduced by [21].

The method implemented here consists of two steps [22]: color normalization and color transformation. The normalization step is applied to make sure that intensity values of the image falls within $[m, 255 - m]$. Note that m, which is set to 10, is a margin added to the standard image range of $[0, 255]$ to lower the risk of exceeding that range after performing the color transformation.

From the original image I, the normalized image I_n is obtained by:

$$I_n = s * I + t \tag{1}$$

where $s = (256 - 2m)/(w_o - b_o)$ and $t = m - b_o$, which are scaling and translation factors. w_o and b_o are the gray-scale values (averaged RGB values) of the reference white and black colors, which are also given by the color checker.

The next step is the color transformation by which the color-calibrated image I_c is obtained:

$$I_c = M * I_n \tag{2}$$

M is a 3x3 matrix calculated as the least-square solution of the transformation of the 24 reference color values (in RGB space) relative to their ground truth values.

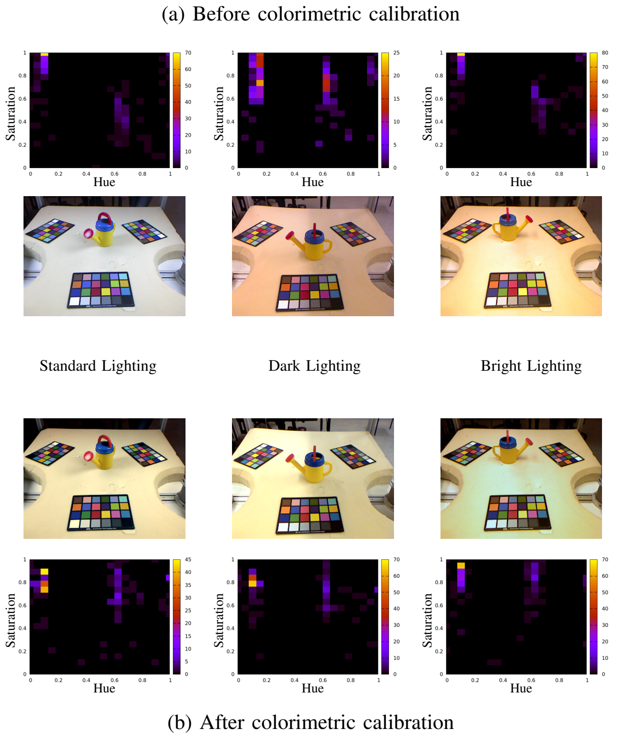


Fig. 3. Colorimetric camera calibration under varying lighting conditions. Object samples (images from one of the Kinect sensors) and their corresponding 2D histogram of hue and saturation are shown for: (a) before applying the calibration and (b) after. The x-rite color checker used to get reference color values is shown at the bottom of each image. This figure should be viewed in color.

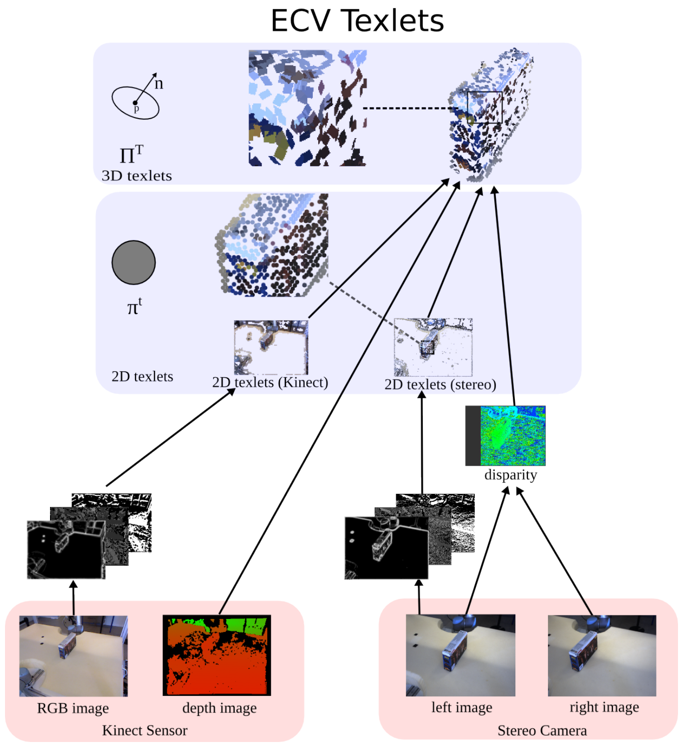


Fig. 4. The hierarchical representation of the texlets. Example images from Kinect and stereo cameras are shown at the bottom. In the middle, 2D texlets are extracted after filtering operations. On top are the extracted 3D texlets from different cameras. This figure is best viewed in color.

D. 3D Texlets

As visual descriptors of objects, 3D texlets are extracted from both stereo and Kinect sensors. Fig. 4 shows the extraction process. 3D texlets (the top level in the figure) represent small, flat local surface patches in the Euclidean space. A 3D texlet is constructed by fitting a plane to a cloud of 3D points surrounding the 3D position that corresponds to the 2D position of a 2D texlet, which is a primitive feature extracted through local filtering of images [23]. The 3D reconstruction is performed using the depth image for Kinect and the dense disparity map (OpenCV implementation of the semi-global block matching algorithm [24]) for the stereo cameras. In the following, we provide a brief description of 3D texlets attributes used in this paper—for full description, the reader is referred to [23].

We define as $\mathcal{T}$ the space of all texlets and the 3D texlet, $\Pi_i^T \in \mathcal{T}$, is formalized as:

$$\Pi_i^T = (\mathbf{p}_i, \mathbf{n}_i, \mathbf{c}_i) \qquad (3)$$

where the index i is used to identify the texlet Π_i^T, $\mathbf{p}_i$ is the texlet's 3D position and $\mathbf{n}_i$ its orientation (given by the normal vector). In addition to the above geometric attributes, the 3D texlets also encode color information in RGB format: $\mathbf{c}_i = (r_i, g_i, b_i)$. When operated in GPU, 3D texlets extraction with Kinect can be achieved with approximately 5 Hz [23].

E. Relations

The 3D texlets introduced in the previous section provide an absolute features (relative to an external reference frame) of objects in the 3D space. One limitation when representing shapes, with e.g., bags of features [13], is that this representation may vary drastically depending on viewpoint. For this reason, we propose to represent objects' shapes as distributions of *relations* between features, that are intrinsically pose-invariant. Pose-invariance is necessary for efficient learning of object classes.

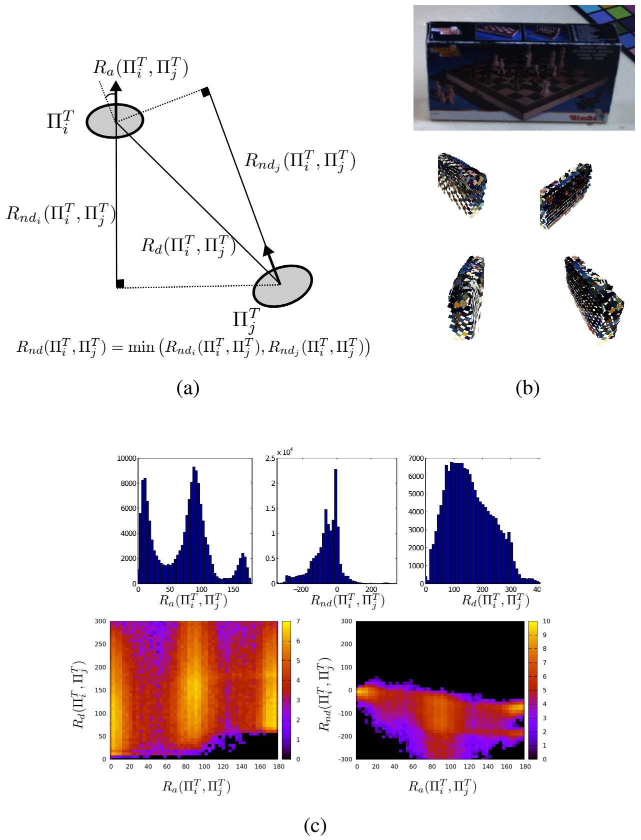
Shape relations are similar to the 3D shape context introduced as local descriptors by [12], however, they are used here as global descriptors of objects. Having combined multiple 3D views of objects allows the global descriptors to be robust and rich representations for fast learning. We also use the term *color relations* to refer to color descriptors, which provide a more robust appearance descriptors compared to the absolute color. This section gives a detailed description of how the different relations are computed.

To describe an object, we compute a set of relations from all pairs of texlets in the object. Formally, a pairwise relation $\mathcal{R}_k$ between texlets is defined as:

$$\mathcal{R}_k : \mathcal{T} \times \mathcal{T} \longrightarrow \mathbb{R} \qquad (4)$$

Hence, a shape described by a set of N texlets $S = \{\Pi_1^T, \Pi_2^T, \ldots, \Pi_N^T\}$ will then be described by $N \times (N-1)$

[6] We use the standard x-rite color checker, see http://xritephoto.com/



(a)

(b)

(c)

Fig. 5. Texlet's shape relations. (a) definition of shape relations between texlet Π_i^T and texlet Π_j^T; Euclidean distance $R_d(\Pi_i^T,\Pi_j^T)$, angle $R_a(\Pi_i^T,\Pi_j^T)$ and normal distance $R_{nd}(\Pi_i^T,\Pi_j^T)$. (b) example of an object and its extracted texlets as seen from different views (c) shape relation histograms of all pairs of texlets extracted from object shown in (b), 1D histograms on top and 2D histograms at the bottom.

values for a given relation. For convenience, we will note the set of those values as $\mathcal{R}_k(S) \in \mathbb{R}^{N\times(N-1)}$, where

$$\mathcal{R}_k(S) = \left\{R_x(\Pi_i^T,\Pi_j^T) : i,j \in [1,N], i \neq j\right\} \tag{5}$$

and $R_x(\Pi_i^T,\Pi_j^T)$ is the inter-texlet relation between Π_i^T and Π_j^T.

One important aspect is that the relation transforms an absolute pose-dependent representation in S into a relative pose-independent one in $\mathcal{R}_k$. For instance, the distance relation $\mathcal{R}_d$ transforms texlets' positions into inter-texlet distances. Because this kind of relations involves pairs of texlets, we refer to it as 'second-order' relations.

In the following, we describe all the texlets relations used in this paper.

Shape relations

The first class of relations that we will consider are *Shape relations*, which are defined to encode the objects' geometric information. This section introduces three shape relations used in this paper. Later, we will investigate which and how to combine those relations for best performance (see Sect. IV-C).

It is important to note that for instance recognition, our shape representation should be *scale-variant*, i.e., object size matters and shall be encoded. Additionally, to characterize the different shape variations, we need to encode the deviation in orientation, i.e., curvature in a global context. Therefore, our set of relations shall address those two aspects. In this paper, we introduce the following relations (illustrated in Fig. 5a):

Angle relation: It is defined as the angle between the two texlets' normals.

$$R_a(\Pi_i^T,\Pi_j^T) = \measuredangle(\mathbf{n}_i,\mathbf{n}_j) \in [0°,180°]$$

The angle relation is important to describe the shape variations. For instance, a flat surface will be dominated by 0° angle relations, whereas a sphere will have a set of relations that are uniformly distributed within the range $(0°,180°]$.

Distance relation: It is defined as the Euclidean distance between two texlets in the 3D space.

$$R_d(\Pi_i^T,\Pi_j^T) = ||\mathbf{p}_i - \mathbf{p}_j||$$

The distance relation describes how texlets are distributed relative to each other. Note that, we don't apply scale normalization to keep the size encoded.

Normal distance relation: The normal distance relation is defined by:

$$R_{nd}(\Pi_i^T,\Pi_j^T) = \min\left(R_{nd_i}(\Pi_i^T,\Pi_j^T), R_{nd_j}(\Pi_i^T,\Pi_j^T)\right)$$

where

$$R_{nd_i}(\Pi_i^T,\Pi_j^T) = (\mathbf{p}_j - \mathbf{p}_i)\cdot\mathbf{n}_i$$

and

$$R_{nd_j}(\Pi_i^T,\Pi_j^T) = (\mathbf{p}_i - \mathbf{p}_j)\cdot\mathbf{n}_j$$

For an object with two parallel surfaces, the normal distance describes the distance between those surfaces, and therefore, explicitly encodes the object's dimensions. Additionally, this relation encodes whether two surfaces are pointing inward (toward each other) or outward; specifically, *positive* value for inward distance and *negative* value for outward. This allows for explicit characterization of certain object properties such as openness and closeness (see Subsect. III-F).

Note that, the two requirements of describing the geometric variation and being scale-variance can be well-fulfilled by combining the angle relation with either the Euclidean distance relation or the normal distance relations.

Color relations

The second class of relation describe the object's appearance using color. The color relations are computed from color channels of HSV and CIELAB (or Lab) spaces. Those two spaces are commonly used for color indexing [25] because they provide a color coding that is more stable under changing lighting conditions than RGB . They both separate the lighting information, luma, from the color information, chroma. More specifically, in HSV, the chroma is represented by the Hue (H) and the Saturation (S) whereas the luma is represented by the value (V). In CIELAB, the luma is the lighting (L) component and the chroma is the a and b components. This allows for the presentation of color with two values, when luma is undesired.

The inter-texlet relation of a certain color channel, c, is computed as:

$$R_c(\Pi_i^T, \Pi_j^T) = \langle c(\Pi_i^T), c(\Pi_j^T) \rangle$$

where $< x >$ denote the average of x. Using the average as color relation maintains the distinctiveness of color as a feature for objects with uniform colors whereas the difference of colors as used in, e.g., [26], would be close to zero. That would mean that homogeneously colored objects of different colors would not be distinguishable. Furthermore, averaging smooths out the noise and hence enhances the color robustness. In practice, experiments on our dataset showed that recognition performance was reduced by nearly 50% when using color difference rather than color average.

For the three color channels of HSV space, the average inter-texlet relations are referred to as $R_h(\Pi_i^T, \Pi_j^T)$, $R_s(\Pi_i^T, \Pi_j^T)$ and $R_v(\Pi_i^T, \Pi_j^T)$. For CIELAB, they are $R_l(\Pi_i^T, \Pi_j^T)$, $R_a(\Pi_i^T, \Pi_j^T)$ and $R_b(\Pi_i^T, \Pi_j^T)$.

The transformation from the RGB space, which is the default space in 3D texlets, is implemented using the standard formulae[7]. Note that $sRGB$ is the RGB standard used by the sensors, hence, the $sRGB$ corresponding white point reference is used to convert to CIELAB space.

F. Multi-dimensional histograms

In the previous section, we introduced a set of relations between texlets. Although individual texlets carry implicit information about objects' shape and appearance, overall statistics over different relations between texlets forming an object do provide pose-invariant and rich description of the objects. This statistical representation is implemented by binning relations in multi-dimensional histograms, which model their distributions as fixed-sized vectors.

For instance, angle and distance relations are mapped into a 2D histogram and color relations formed from the three color channels are mapped into a 3D histogram. In the following, we will show that such a representation of objects is naturally pose-invariant.

For a set of D relations, denoted as $V = \{\mathcal{R}_{x1}, .., \mathcal{R}_{xD}\}$, the D-dimensional histogram is defined as:

$$H(V, b) = \{h_1(V), ..., h_{b^D}(V)\} \qquad (6)$$

where b is the number of bins that is, for simplicity, kept constant along all dimensions (relations) and $h_i(V)$ is the number of relations that fall jointly within the boundary of the ith bin. This means that the total number of the bins in this multi-dimensional histogram (i.e., the size of the corresponding feature vector) is equal to b^D. The optimal value of b is experimentally determined for all kind of relations in different ways of grouping (see Sect. IV-C). All bins of the multi-dimensional histograms can then be used as a fixed size feature vector, f, describing the object's shape and appearance.

Fig. 6 shows 2D histograms for different objects. In this figure (a) and (b) represent the same box with two different poses and appearances; (c) is a similar box, but with an empty cavity in the front side; and (d) is a cylindrical box. First, note that the shape histograms in (a) and (b) are very similar, despite the object being in a different pose. This demonstrates the invariance of the relation statistics as a feature descriptor. For those two objects, the 2D histograms of shape (on the left) illustrate characteristics of the object's shape: the peak visible for normal distance of zero and angle of zero encodes all coplanar texlets. Then, two peaks are visible for angle of 180 degrees and normal distance of -150 and -270 that correspond to the parallel sides of the box. Finally, the area around 90 degrees correspond to orthogonal surfaces.

Second, in (c) we can see the representation for an open box. In this case, the color histogram (right) is similar to (b), but it also shows additional peaks for the inside color. In the shape histogram, we also see additional peaks illustrating the parallel surfaces from the inside and outside of the box. For the cylindric object in (d), the shape histogram as well the appearance histogram are significantly different from the rest.

In summary, the above examples demonstrates three characteristics of the shape relations: pose-invariance, distinctiveness and interpretability.

G. Histogram processing

In previous sections, we showed how histograms of relations provide a rich description of objects. In such a high level representation, reducing noise will enhance recognition. The noise is a result of error propagated from lower processes such as 3D reconstruction, relative camera calibration, texlet sampling, uncompensated variation in color, and histogram binning.

We use Gaussian smoothing filter to perform noise reduction by convolving the histogram with a Gaussian (normal distribution) function. Gaussian filter is a low-pass filter that reduces the noise and only attenuates the high-frequency components because it does not have a sharp cut-off frequency. The filter is widely used in image processing applications (e.g., Canny edge detector [27]) where the information is contained in the high-frequency components. This also applies to our histograms – shape and color information are high-frequency. Let $\acute{H}(V, b)$ be the histogram after smoothing, which is computed by:

$$\acute{H}(V, b) = H(V, b) * K(\sigma) \qquad (7)$$

where $K(\sigma)$ is a D-dimensional Gaussian kernel and σ is its standard deviation, which will be chosen empirically in Sect. IV-C. In implementation, we make use of the separability property of the Gaussian kernel. Therefore, the D-dimensional convolution is performed by a series of D consecutive convolutions using 1D kernel. The kernel size is determined using the 3-sigma rule ($k = 10/3(2\sigma - 1)$), which implies that the kernel covers 99.7% of the Gaussian function.

[7] See e.g.,http://brucelindbloom.com

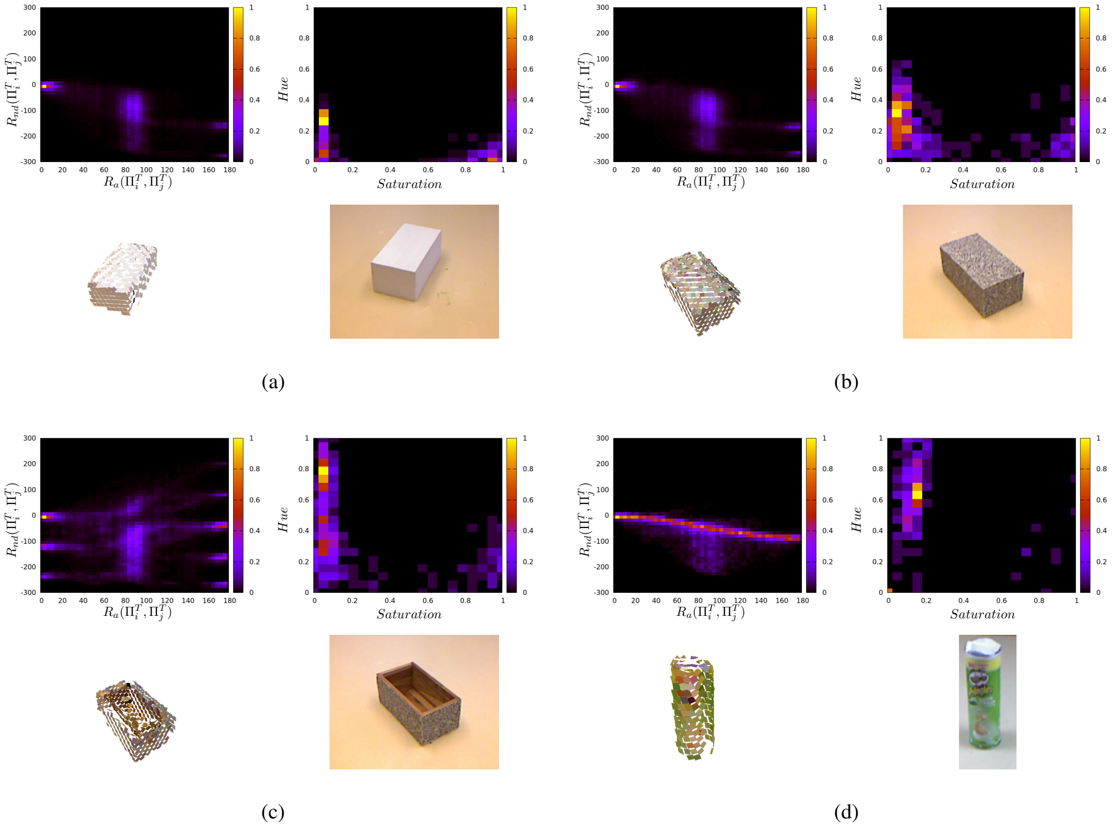


Fig. 6. Four different scene configurations and corresponding histograms. The histogram blocks for each scene consists of the following components: (top row, left) 2D histogram of angle $R_a(\Pi_i^T, \Pi_j^T)$ and normal distance $R_{nd}(\Pi_i^T, \Pi_j^T)$ for all possible pairs of texlets. (top row, right) 2D appearance histogram representing the hue (H) and the saturation color information. (bottom row, right) overview of the object in the scene. (bottom row, left) the extracted 3D texlets of the object.

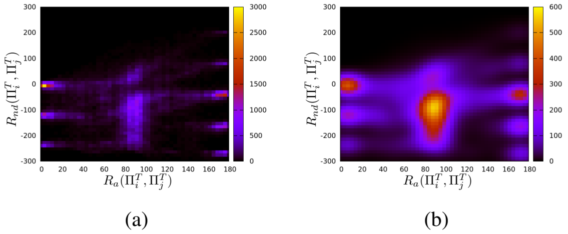


Fig. 7. Histogram filtering. (a) the original histogram (b) the histogram after filtering with $\sigma = 2$.

H. Classification (Random forests)

The quality and invariance properties of the histogram representation presented in the previous section makes it attractive for the purpose of object recognition. Supervised classification is a field that is well explored in machine learning (e.g., [28], [29]). In this work, we make use of Random Forest classification [4], [5]. The reasons for this choice are multiple: first, RF can be trained efficiently and are very fast at classification time, even for large input dimensions; second, RF are intrinsically multi-class allowing for an efficient learning in contrast to 1 vs. all approaches; third, RF have shown to reach very high level of performance on a variety of tasks (notably [30], [31]); finally, RFs effectively perform a form of dimension selection and which makes the resulting models interpretable.

Random forests learn a collection of randomized decision trees from different random subsets of the available training data, in a manner similar to *Bagging* [32].

Formally, if we consider a dataset $D = (f_j, y_j)_{j \in [1..|D|]}$, where f_j is an observation represented by the feature vector f presented in Subsect. III-F and $y_j \in [1..C]$ is a class label and $|D|$ denotes the number of samples in D, then we draw M random subsets $D_i \subset D$, $\forall i \in [1..M]$ from the data ($M = 100$ here) and train a population of M decision trees $P = T_{i, i \in [1..M]}$ such that T_i is trained from the subset D_i. Typically, the subsets D_i are drawn randomly such that $|D_i| = \gamma|D|$ (we used a common value of $\gamma = 0.5$).

From each subset D_i, we train a Randomized Classification Tree (RCT). RCT are binary trees, where each node n splits the input space (and thereby the dataset such that $D_l \cup D_r = D_n$) recursively in order to maximize class purity in all partitions and sending the samples that fall on each side of the partition to each child node. The recursion stops when a node receives too few samples to split ($|D_n| < 5$ here) or reaches a maximum depth ($\mathrm{depth}(n) > 10$ here)—such nodes are called leaf nodes and label the corresponding region according to the majority label in the available samples.

The split operation is traditionally done along a hyperplane, by applying a threshold operation to one input dimension. The randomization of the decision trees is done by selecting randomly a subset of input dimensions (computed to be the first integer less than $log_2|f| + 1$, [5]) for each non-leaf node and optimizing amongst those the dimension and threshold defining the split which minimizes all partitions' class impurity, using the so–called Gini coefficient $G(D_n)$:

$$G(D_n) = 1 - \sum_{k=1}^{C} \left(\sum_{j=1}^{|D_n|} \frac{\mathbf{I}_k(y_j)}{|D_n|} \right)^2 , \tag{8}$$

where $\mathbf{I}_k(y_j)$ is an indicator function that returns 1 if $y_j = k$ and 0 otherwise.

Finally, the RF prediction $P(f)$ for an input vector is obtained by calculating the class with the largest amount of votes amongst all RCTs T_i, i.e.,:

$$P(f) = \arg \max_{k \in [1..C]} \sum_{i \in [1..M]} \mathbf{I}_k(T_i(f)) \tag{9}$$

hlwhereas the associated *confidence* is computed as the ratio of the number of votes (of the predicted class) to the number of RCTs.

The hierarchical greedy search for splits allows for a high performance classification, while the randomization and redundancy provided by the bagging reduces the model's overfitting, increasing generalization and robustness.

IV. Dataset and Experiments

In this section, we present the benchmark dataset and the different experiments performed to evaluate the system. First, the multi-view object dataset is introduced in Subsect. IV-A. This is followed by describing how the experiments are set up in Subsect. IV-B. In Subsect. IV-C, we investigate how to form the optimal description of an object by separately considering the color and shape representations. Using multiple camera views versus single view is compared in Subsect. IV-D. In Subsect. IV-E, we show the performance obtained through Kinect data in comparison with stereo data. In Subsect. IV-F, we do error analysis by discussing the cases in which the system performs relatively low.

A. Multi-view object dataset

To benchmark our system, a dataset of 100 objects was created[8](see Fig. 8). The selection of objects cover a wide range including industrial and household objects, some of them taken from the KIT dataset [33]. The dataset contains RGBD and stereo images from the three Kinects and stereo pairs presented in Subsect. III-B, along with the relative transformations of the sensors (calibrated). For each sample, we extract 3D texlets (as discussed in Sect. III-D) from all views. Texlets from different views are then combined (in 3D space) using the camera transformations. This allows for having a rather complete 3D visual representation of objects (see Fig. 5b).

There is a total of 30 different samples (random poses) for each object captured under three defined lighting conditions (see Fig. 2a): 'standard', 'dark' and 'bright' with 10 samples each. The variation in lightning is created to test the robustness of the system in light-changing conditions and to study the impact of the colorimetric calibration. Fig. 3 shows samples of the different lighting conditions. Objects were selected such that the set has objects with the same shape and different appearances and vice versa (see Fig. 8). The reason for this is to test the use of shape and color both individually and in combination.

B. Experiment setup

In the following experiments, unless otherwise specified, the 3D texlets from three Kinects (colorimetrically calibrated) are used. In each experiment, the dataset is divided into training and test subsets. The test subset is taken from one lighting condition (10 samples per object) whereas the training subset is taken from the other two (20 samples per object). The test set is used to evaluate the system performance in terms of recognition accuracy, which is defined as the trace mean of the confusion matrix.

To quantify the robustness of color information associated to the texlets, the experiment is executed in three different modes. For each mode, the test subset is taken from a different lighting condition and the experiment is iterated 5 times where the RF is differently seeded each time. This results in 15 iterations from which the average accuracy and the standard deviation are computed.

Here, we want to point out that a big advantage of our ~~our~~ approach is that good recognition performance is already possible with very few object instances stored (see section IV-D) due to the high degree of pose invariance of the representation as well as the color normalization procedure. This allows for a fast teaching of objects by putting them into the field of view of the camera system and record data for very few standard poses (e.g., two for a cylindric object corresponding to standing and lying). This fast teaching is particularly important in an industrial context.

C. Optimal representation for color and shape

In this section, various experiments have been conducted to find the best combination of color and shape relations and to determine histogram and filtering settings. The process has been performed for color and shape relations separately, such that they can later be combined in one representation. Whenever color is involved, the use of the colorimetric calibration is also evaluated. The results presented in this section address the following aspects:

Set of relations: Here, we aim at selecting the best set of relations encoding shape and color, from the ones defined in Subsect. III-E.

[8] http://caro.sdu.dk/sdu-dataset (we will make it available by the final submission of this manuscript)

Fig. 8. Multi-view dataset of 100 objects shown in thumbnails. The set contains RGBD and stereo images: 30 samples each object from three camera views. The data were captured under three lighting conditions. The dataset is available at http://caro.sdu.dk/sdu-dataset (we will make it available by the final submission of this manuscript).

Histogram binning: To determine the optimal bin size of histograms. For simplicity, the bin size is fixed across dimensions.
Filtering: To determine the value of σ (Gaussian filtering of histograms) that yields the best recognition.
Relational dimensionality: To determine the construction of relations into ND histograms, i.e., the best composition of the feature vector f defined in Subsect. III-E. For instance, three relations can be arranged in three 1D histograms, two 2D histograms or one 3D histogram. The optimal color and shape representations are separately determined by investigating the above aspects. The overall object representation is then defined by the combination of the two representations.

Color

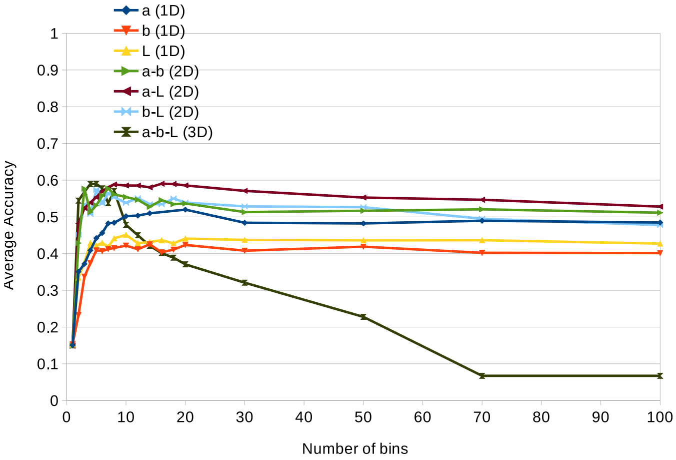


Fig. 9. Color histograms binning. The histogram binning of CIELAB color relations: a, b and L are 1D histograms of the color space components; a-b, a-L and b-L are 2D histograms; and a-b-L is a 3D histogram.

Fig. 9 shows the classification accuracy over varying number of bins using different color relations derived from CIELAB space. Generally, the figure shows that the performance increases to a maximum value before it starts to decrease again. The decrease is steeper for histograms of higher dimensions – this is particularly clear for the 3D histogram. This can be interpreted as a result of data sparsity in feature space, which is exponentially proportional to the number of dimensions. Moreover, the higher the number of bins, the higher the number of features involved in learning (see Sect. III-F). This makes learning slower and more prone to over-fitting.

From the figure, we find that the optimal number of bins is 10, 6 and 4 for 1D, 2D and 3D histograms, respectively. The number of bins corresponds to a resolution of 10% of the color space in 1D histograms, 16.7% in 2D histograms and 25% in 3D histograms.

Based on the optiaml bin numbers, we experimented the effect of filtering under varying values of σ. We found that $\sigma = 1$ yields the highest performance. This value of σ corresponds to 5% of the color space in 1D histograms, 8% in 2D histograms and 12.5% in 3D histograms.

In Fig. 10, the HSV and CIELAB color spaces are compared in terms of the system classification accuracy when color relations are used. In this figure, color relations from the different components binned in ND histograms are shown. The figure also demonstrates the effect of the colorimetric calibration in each case. We can observe a significant improvement with calibration in all cases except for the L component of the CIELAB and the *value* component of the HSV. Although the Kinect sensor automatically performs exposure adjustment resulting in stabilizing luma components, which L and *value* represent, the result shows that luma is not a strong feature for recognition under changing lighting conditions. The figure shows that, in all cases, the colorimetric calibration accounts for smaller standard deviation, i.e., higher stability. It also shows that CIELAB outperforms HSV as a color space when color is used for recognition.

In Fig. 11, we show the classification accuracy when the three components of CIELAB arranged in different dimensionalities: three 1D histograms, two 2D histograms and one 3D histogram, hence it shows all the possible arrangements in which the three color components can be combined. In determining the overall color representation, we find that 1D histograms (1D histograms of L relations, a relations and b relations) slightly outperform the 2D histograms. Additionally,

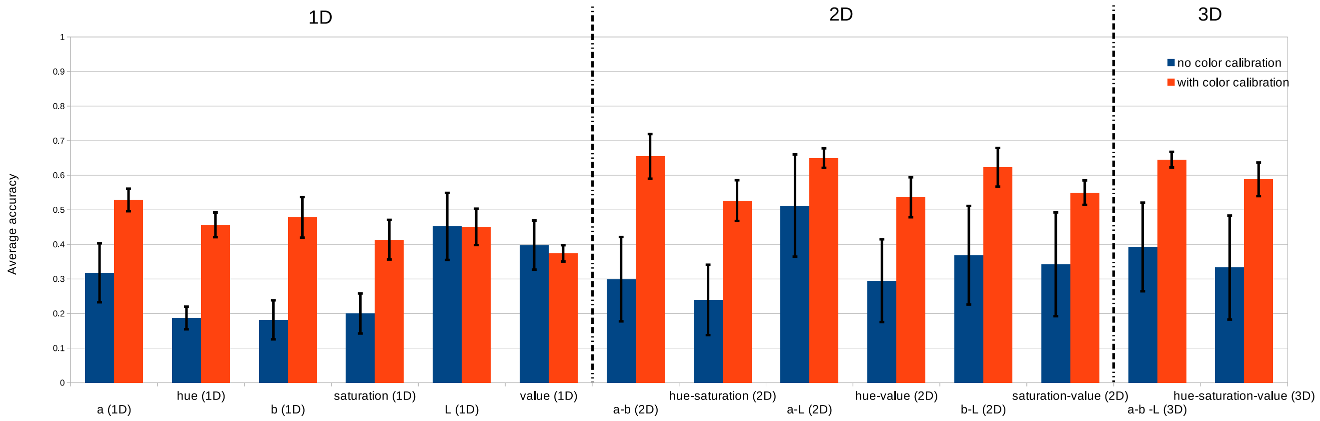


Fig. 10. CIELAB Vs HSV. a, b and L are 1D histograms of CIELAB components; a-b, a-L and b-L are 2D histograms; and a-b-L is a 3D histogram. Hue, Saturation and Value are 1D histograms of HSV components; Hue-Saturation, Hue-Value and Saturation-Value are 2D histograms; and Hue-Saturation-Value is a 3D histogram.

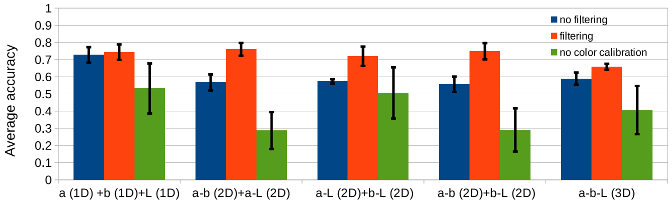


Fig. 11. Color relation dimensionality. a+b+L is the combined 1D histograms of CIELAB components; a-b+a-L, a-L+b-L and a-b+b-L are two combined 2D histograms each; and a-b-L is one 3D histogram

the figure also emphasizes the advantage of filtering and the colorimetric calibration.

Shape

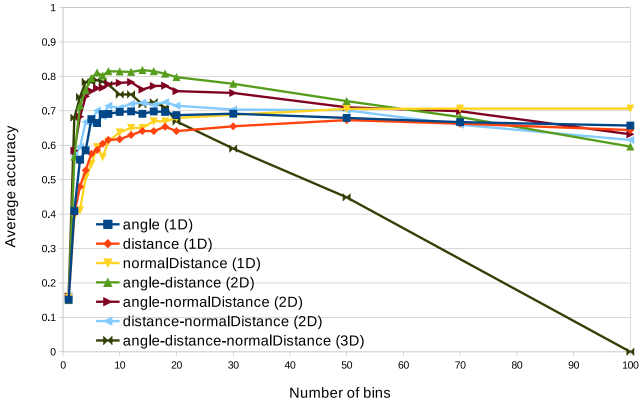


Fig. 12. Shape histograms binning. The histogram binning of the shape relations: Angle, Distance and NormalDistance are 1D histograms; Angle-Distance, Angle-NormalDistance and Distance-NormalDistance are 2D histograms; and Angle-Distance-NormalDistance is a 3D histogram.

Similar to color, we first aim at determining the optimal number of bins for the different shape relations discussed in Sect. III-E as shown in Fig. 12. We can see the same pattern occurring: the performance reaches a maximum value before it starts to decrease and that it has a steeper slope for higher dimensions. From the figure, we find that the optimal number of bins is 50 for distance and 19 for angle in 1D histograms, 12 in 2D histograms and 8 in 3D histograms. The number of bins corresponds to a resolution of 2% of the shape relations spaces in 1D histograms, 8.3% in 2D histograms and 12.5% in 3D histograms. Note that the distance ranges from 0 to $300mm$ and the angle ranges from 0 to 180°.

Based on the optiaml bin numbers, we experimented the effect of filtering under varying values of σ. We found that $\sigma = 0.5$ yields the highest performance. This value of σ corresponds to 1.2% of the shape spaces in 1D histograms, 2.1% in 2D histograms and 3% in 3D histograms.

When the shape relations arranged in different dimensionalities, we found that the 2D histogram of angle and distance yields the best performance. Moreoever, as opposed to color, we found that filtering in shape relations does not achieve significant improvement.

Combined shape and color

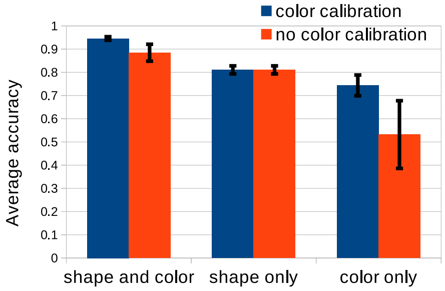


Fig. 13. The optimal representations of color and shape separately and combined.

Based on the above findings, we show how the system performs when the optimal color and shape representations are combined. This is demonstrated in Fig. 13 and also compared with the separate representations of color and shape. The figure shows that a classification accuracy of 94% is achieved, which is significantly higher than 81% for shape alone and 74% for color alone. We can also see that the colorimetric calibration contributes with improving the accuracy as well as the stability (i.e., smaller standard deviation).

D. Performance depending of number of views and samples

In this experiment, we show the system performance while increasing the number of training samples. For each experiment, the number of samples per class is fixed to 10 samples in the test subset and changed from 1 to 20 in the training subset. Fig. 14 shows three learning curves for three cases: three views, two views and one view. Note that, in contrary to previous experiments, for both subsets, the samples are randomly selected across all lighting conditions. This explains the slightly higher performance for three views with 20 samples per class in training (96% compared with 94% as in Fig. 13).

The figure highlights important features of the our system. First, the learning efficiency by which high performance is achieved with a few training samples. We can observe that with already 1 sample per class we get 60 % and above 90 % with 5 samples. Secondly, the figure shows the advantage of having multiple views. This is evident in terms of performance (about 17% improvement compared with one view, 5% compared with two views). It is also evident by obtaining faster learning with more views, i.e., less number of samples is needed to reach the 'steady-state' of accuracy (it is about 18, 15 and 10 for one, two and three views respectively).

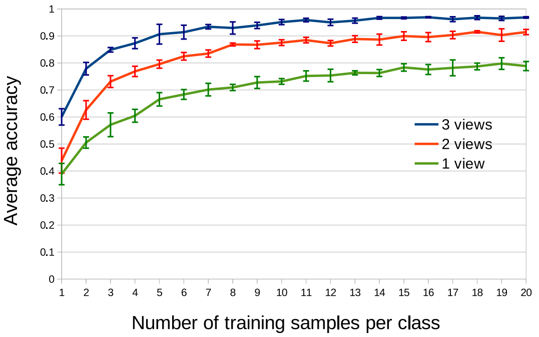


Fig. 14. Learning curve for object instance recognition. Three cases are compared depending on the number of camera views: singe view, two views and three views.

E. Kinect vs stereo cameras

Figure Fig. 15 shows the classification accuracy of the system when the stereo cameras are used compared with the Kinect sensors. The figure shows that the system performed significantly better with Kinect data (26% higher) on our dataset. Contrary to Kinect data extraction, dense stereo algorithms generally fails on non-textured objects. Having many objects in the dataset that fall in this category explains the lower performance of the stereo data. Non-textured objects are widely available especially in industrial platforms and that limits the reliability of the stereo sensory data.

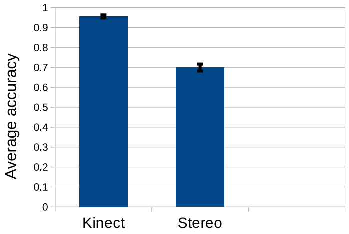


Fig. 15. System performance on stereo versus Kinect data.

F. Misclassification analysis

The maximum classification accuracy the system reaches is 96% (Fig. 14). In Fig. 16, the 3 objects with the lowest classification accuracy are presented together with their top confusing objects. The figure is derived from the average confusion matrices computed within the same experiment discussed in subsection IV-D.

The confused objects, as shown in the figure, are very similar in shape or color. The two objects on the left and the center are confused with objects that have the same shape, which suggests that the system fails to detect differences in their color representations. The object on the right is confused with an object with the same color and a only slightly different shape. Given that the two objects are relatively small, such geometric differences are beyond the limit of the sensor (Kinect in this case) to extract any distinctive 3D information.

The system accuracy discussed above considers only the RF top prediction, i.e., the prediction with the highest confidence (or the majority of tree votes). If we allows for predictions of confidence values that are above a certain threshold, we will obtain, instead of single prediction, a list of recognition candidates per test instance. In order to find the accuracy limit the system can reach by possibly including a process capable of finding the correct prediction from this list. To do this, a test instance is considered correctly recognized if it is in the list. The confidence threshold is set to 25% allowing for a maximum of 4 candidates. By applying the same settings as in Subsect. IV-D, we reach an average accuracy of 99.76% with a standard deviation of $6{\times}10^{-3}$.

V. Application for Pose Estimation

We have tested our system in the more application-oriented scenario of free-form recognition and full 6D pose estimation. In this application, we assume a typical tabletop setting where multiple objects are observed in a scene (see Fig. 18). The task is to perform instance recognition and pose estimation of all objects present for further processing, e.g. robotic manipulation. To facilitate the use of our representation in the recognition process, we assume spatial separability of the objects, which allows us to preprocess and segment the scene

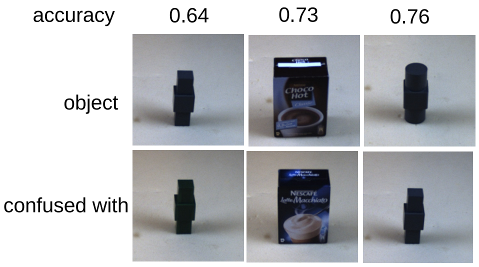


Fig. 16. The three least classified objects and the objects they are confused with. Note that the objects on the left have different color (top: dark gray, below: dark green). The thumbnails are resized for better visualization and they don't necessarily reflect their actual relative sizes.

(see Subsect. III-A) and then recognize all the clusters. Note that such a separation is straightforward to achieve in an industrial setting, by, e.g., any feeders.

Our algorithm for recognition and pose estimation works as follows:

1) *Cluster recognition:* the RF classifier is now run on each of the clusters separately. If the classifier returns a prediction confidence below 0.25 for a cluster, the cluster is rejected as an unknown object.
2) *Nearest training instance search:* the RF classifier is designed in such a way that it generalizes over the training instances for identification of an object in novel views. The RF output of a cluster is thus the ID of the object producing the highest prediction confidence. For pose estimation, however, we wish to perform a 3D alignment between the identified object and the scene cluster. To this end, we do a search for the concrete three view training instance of the object showing the highest degree of similarity with the cluster and use this model to compute the relative pose. This information is not available in the RF output, so we perform a linear search within the training set for the nearest matching view using the global histogram descriptors.
3) *Pose estimation:* the recognized object is now aligned with the scene cluster using the identified training instance. Here we use an optimized RANSAC-based algorithm presented in our prior work [34]. This algorithm injects a prerejection step to quickly discard samples that are unlikely to produce a correct alignment, making the search for the pose much faster. The best pose is determined by the number supporting inliers, given by the number of aligned object points that lie within 5 mm of the nearest scene point. The output pose of the RANSAC algorithm is finally refined using the ICP algorithm [35] to get a more accurate pose.

The above procedure is repeated for all clusters in the scene for which the RF classifier returns a high enough confidence. A block diagram is shown in Fig. 17 and the procedure is used as a direct addition to the recognition procedure in Fig. 2. The whole pose estimation process for each object, including pose refinement, takes on average less than 500 ms, due to the prerejective nature of the modified RANSAC algorithm.

In Fig. 18, we show pose estimation results for several different scenes of varying difficulty. During these tests, we experienced a very high amount of accuracy, as long as the objects were clearly visible in the scene

VI. Conclusion

We presented an object instance recognition system for an industrial work-cell with multiple vision sensors. Our system represents objects with viewpoint invariant 3D shape features as well as robust color features. The system was evaluated on a dataset of 100 objects recorded under three lighting conditions.

The results show that our system is able to achieve high performance (in terms of classification accuracy) with a few training samples. The results also shows that the system performance using multi-view representation of objects, i.e., combined representations of multiple cameras, is significantly higher compared to single view. Regarding color encoding, the result shows that color normalization, which aims at compensating for variation in lighting, enhances the performance. Therefore, the use of multi-view object representation for shape combined with applying color normalization is crucial for a reliable recognition system operating in this set-up. This high reliability allows for using the system to trigger other processes such as pose estimation, which we have also demonstrated in several complex scenes.

Acknowledgment

This work has been funded by the EU project Xperience (FP7-ICT-270273) and the project CARMEN funded by the The Danish Council for Strategic Research.

References

[1] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2009 (VOC2009)," Summary presentation at the 2009 PASCAL VOC workshop, 10 2009.

[2] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 1817–1824.

[3] T. R. Savarimuthu, A. G. Buch, Y. Yang, W. Mustafa, S. Haller, J. Papon, D. M. nez, and E. E. Aksoy, "Manipulation monitoring and robot intervention in complex manipulation sequences," in *Workshop on Robotic Monitoring at the Robotics: Science and Systems Conference (RSS)*, 2014.

[4] Y. Amit and D. G. Y, "Shape quantization and recognition with randomized trees," *Neural Computation*, vol. 9, pp. 1545–1588, 1997.

[5] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[6] W. Mustafa, N. Pugeault, and N. Krüger, "Multi-view object recognition using view-point invariant shape relations and appearance information," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[7] M. Everingham, A. Zisserman, C. K. I. Williams, L. van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. D. R. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. S. Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, and J. Zhang, "The 2005 PASCAL Visual Object Classes Challenge," in *Pascal Challenges Workshop*, ser. LNAI. Springer, 2006, vol. 3944, pp. 117–176.

[8] N. Pinto, J. J. DiCarlo, and D. D. Cox, "How far can you get with a modern face recognition test set using only simple features?" 2009.

[9] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 999–1006.

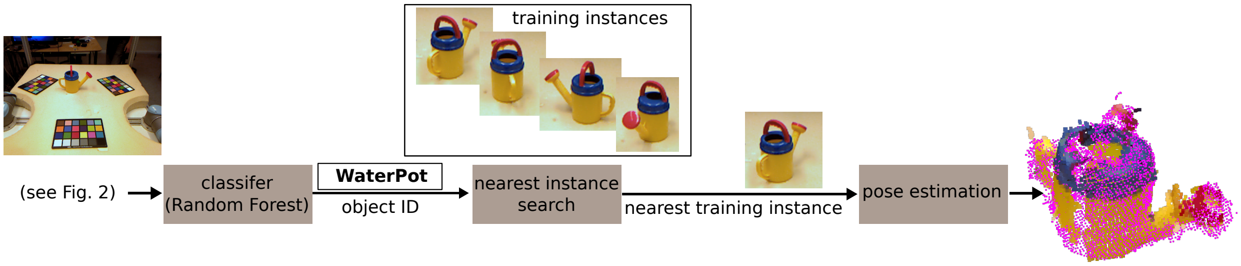


Fig. 17. Object instance recognition and pose estimation for a single object cluster: extended block diagram from Fig. 2, with the aligned training instance shown in purple. See text for details.

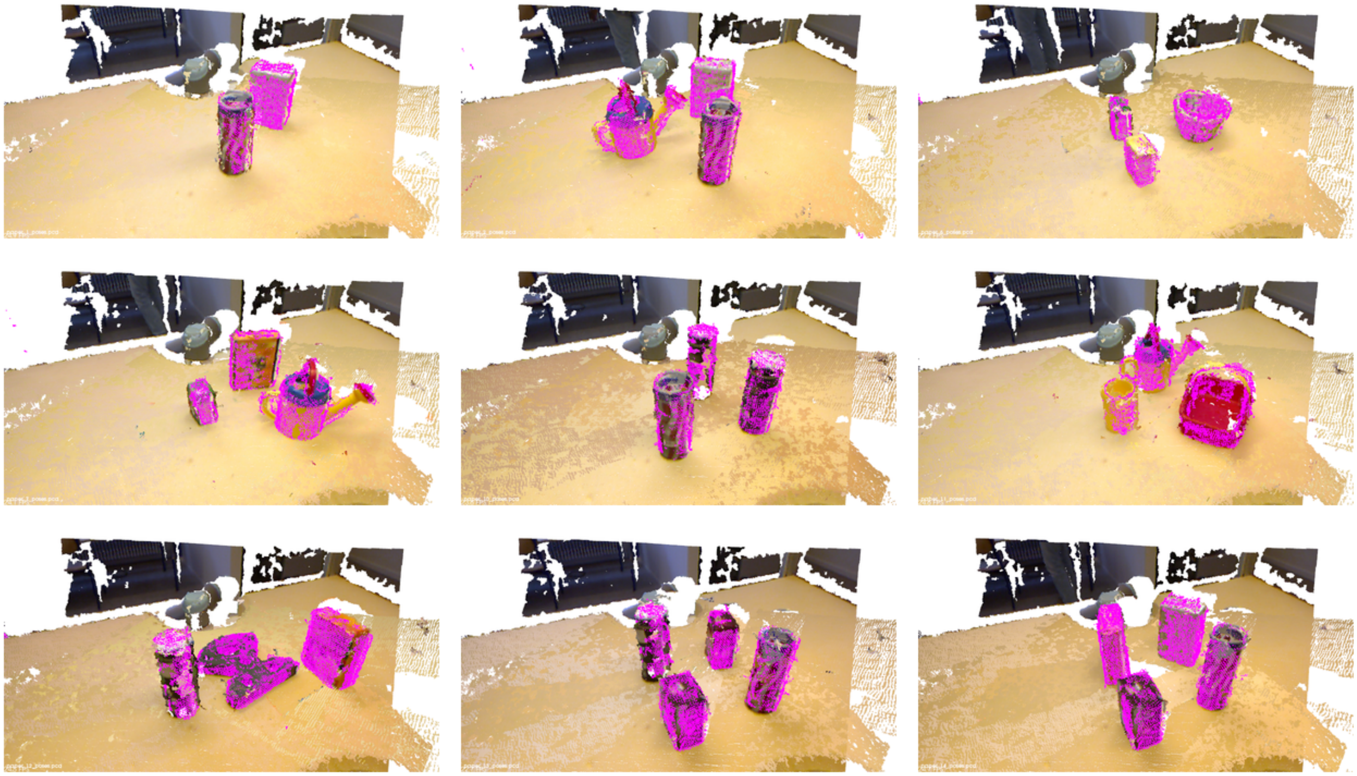


Fig. 18. Object instance recognition and pose estimation results, aligned models overlaid in purple. In all cases, all objects are correctly recognized.

[10] D. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999, pp. 1150 –1157 vol.2.

[11] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 4, pp. 509–522, Apr. 2002. [Online]. Available: http://dx.doi.org/10.1109/34.993558

[12] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, May 2004.

[13] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[14] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 994–1000. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.254

[15] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations (ICLR 2014)*. CBLS, April 2014. [Online]. Available: http://openreview.net/document/d332e77d-459a-4af8-b3ed-55ba

[16] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," *arXiv preprint arXiv:1403.6382*, 2014.

[17] A. M. Pinto, L. F. Rocha, and A. P. Moreira, "Object recognition using laser range finder and machine learning techniques," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 12 – 22, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0736584512000798

[18] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[19] B. Nemec, F. Abu-Dakka, J. Rytz, T. Savarimuthu, B. Ridge, N. Krger, H. Petersen, J. Jouffroy, and A. Ude, "Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile," in *Advanced Robotics (ICAR), 2013 16th International Conference*, 2013.

[20] A. Maimone and H. Fuchs, "Reducing interference between multiple structured light depth sensors using motion," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, March 2012, pp. 51–54.

[21] C. S. McCamy, H. Marcus, and J. Davidson, “A color-rendition chart,” *J. App. Photog. Eng*, vol. 2, no. 3, pp. 95–99, 1976.

[22] Y. P. Touati, “Image color calibration using a color calibration rig,” University of Southern Denmark, Tech. Rep., 2010.

[23] S. M. Olesen, S. Lyder, D. Kraft, N. Krüger, and J. B. Jessen, “Real-time extraction of surface patches with associated uncertainties by means of kinect cameras,” *Journal of Real-Time Image Processing*, pp. 1–14, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11554-012-0261-x

[24] H. Hirschmller, “Accurate and efficient stereo processing by semi-global matching and mutual information,” in *In Proc. CVRP*. IEEE Computer Society, 2005, pp. 807–814.

[25] M. Swain and D. Ballard, “Color indexing,” *International Journal of Computer Vision (IJCV)*, vol. 7, no. 1, pp. 11–32, 1991.

[26] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, “Evaluating color descriptors for object and scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010. [Online]. Available: http://www.science.uva.nl/research/publications/2010/vandeSandeTPAMI2010

[27] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 8, no. 6, pp. 679 – 698, 1986.

[28] C. Cortes and V. Vapnik, “Support-vector networks,” in *Machine Learning*, 1995, pp. 273–297.

[29] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[30] J. Shotton, M. Johnson, and R. Cipolla, “Semantic texton forests for image categorization and segmentation,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, june 2008, pp. 1 –8.

[31] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, june 2011, pp. 1297 –1304.

[32] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996. [Online]. Available: http://dx.doi.org/10.1023/A:1018054314350

[33] A. Kasper, Z. Xue, and R. Dillmann, “The kit object models database: An object model database for object recognition, localization and manipulation in service robotics,” *International Journal of Robotics Research (IJRR)*, vol. 31, no. 8, pp. 927–934, 2012.

[34] A. G. Buch, D. Kraft, J.-K. Kamarainen, H. G. Petersen, and N. Kruger, “Pose estimation using local structure-specific shape and appearance context,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2080–2087.

[35] P. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, Feb 1992.