Integration of Probabilistic Pose Estimates from Multiple Views

Özgür Erkent(✉), Dadhichi Shukla, and Justus Piater

Institute of Computer Science, University of Innsbruck, Innsbruck, Austria
{ozgur.erkent,Dadhichi.Shukla,Justus.Piater}@uibk.ac.at

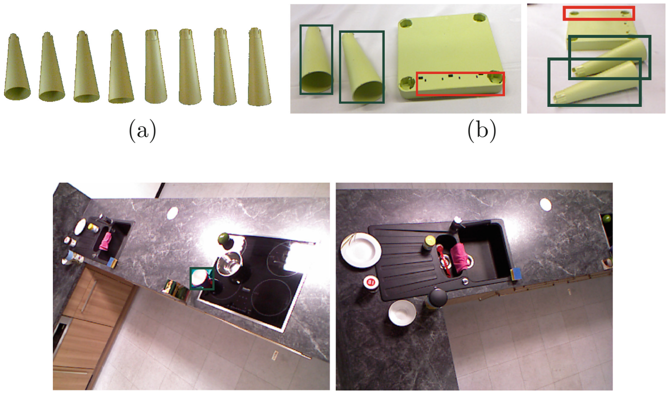
Abstract. We propose an approach to multi-view object detection and pose estimation that considers combinations of single-view estimates. It can be used with most existing single-view pose estimation systems, and can produce improved results even if the individual pose estimates are incoherent. The method is introduced in the context of an existing, probabilistic, view-based detection and pose estimation method (PAPE), which we here extend to incorporate diverse attributes of the scene. We tested the multiview approach with RGB-D cameras in different environments containing several cluttered test scenes and various textured and textureless objects. The results show that the accuracies of object detection and pose estimation increase significantly over single-view PAPE and over other multiple-view integration methods.

Keywords: Pose estimation · Object recognition · Multiple cameras

1 Introduction

Detection and pose estimation of textureless objects are well-studied challenges in robot vision. However, there are still problems that need to be solved. One of the problems is that the estimated pose can be ambiguous due to the ambiguity in the detected shape of the object [22] as shown in Fig. 1a. When a probabilistic, appearance-based pose-estimation method is used, it can be difficult to determine the viewing angle of the object due to similar appearances from the observed views. Another problem is due to the presence of outliers [9] (Fig. 1b). One of the solutions to overcome these difficulties is to observe the scene with multiple cameras. To use multiple attributes of the scene would also improve the pose estimation performance. In this paper, we introduce an approach that uses RGB-D images from different viewpoints to overcome these difficulties. Multi-view integration can face difficult problems when the objects are occluded or totally unseen in one of the views as shown in Fig. 1c. Another difficulty can arise when the sensor information is incomplete or noisy. Noise or incompleteness may even result from interference between multiple RGB-D cameras as shown in Fig. 2. Therefore, we consider the integration of information from multiple RGB-D cameras and pose estimation in the presence of noisy or incomplete data as a coupled problem.

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 154–170, 2016.
DOI: 10.1007/978-3-319-46478-7_10



(a) (b)

(c)

Fig. 1. Some of the problems that can be solved with integration of multi-view pose estimations. (a) Ambiguities in the pose of an object; (b) Correct pose estimates are shown with green bounding boxes in two views. Outliers, which are shown with red, are eliminated after integration; (c) The cup is not visible in the right view. The integration method is capable of finding the object even if it is not visible in all of the views. The images are taken from the MPII Multi-Kinect Dataset [20]. (Color figure online)

For each view, possible 6DoF (3DoF in translation and 3 DoF in rotation) poses of the object are estimated with a probabilistic, appearance-based method which can combine multiple features for recognition. Pose estimates from all of the views are integrated while allowing for absence of a correct estimation from some of the views. Absence of a correct estimation can occur due to various reasons including partial or entire occlusion of the object, unobservability of the object within the limits of the sensor, or a false pose estimate. After integration, all of the integrated pose estimates are associated with a probability value, and the candidate with the highest score is selected as the final estimate.

We use a probabilistic, appearance-based method to detect and estimate the pose of the object from a single view. We introduce an approach to combine different attributes of the scene, e.g., edge orientations, depth values, surface normals, and color. Combining multiple attributes of the scene can increase the performance of recognition in cluttered environments. In the presence of noisy or incomplete data, a “probability of absence” parameter is used as explained in Sect. 2.1.

To summarize, our work makes two main contributions:

- An approach that integrates the pose estimates in 6DoF from multiple views even in the absence of a correct estimation in some of the views.
- A method to combine the different appearance-based attributes in the presence of noisy or incomplete data.

In Sect. 1.1, we review related work. In Sect. 2, we explain how to combine multiple attributes in the probabilistic, appearance-based pose estimation

(PAPE) method. In Sect. 3, we explain our approach to object recognition in a multiview camera setup. The proposed algorithm is evaluated in Sect. 4, and Sect. 5 concludes the paper with a brief summary.

1.1 Related Work

There have been several studies on integrating information from multiple cameras to increase the accuracy of detection and pose estimation of objects. However, only a few of them are interested in the specific task of object detection and pose estimation. For example, "KinectFusion", which is developed by Izadi et al. [12], mainly deals with the problem of scene reconstruction.

Some studies try to find corresponding features in between images. For example, Yang et al. [26] use complex descriptors (SIFT [14]) in sensor networks to detect objects with texture; such methods are not suitable for textureless objects. In another study, Aldoma et al. [1] capture multiple RGB-D images of the same scene from different viewpoints. They reconstruct the scene and transform the hypothesis obtained in each single view into the reconstructed scene. There is no interference noise of multiple Kinects since only a single camera is used. Mustafa et al. [15], compute 3D descriptors in the reconstructed 3D scene, which requires distinctive features and reconstructed 3D data of the scene.

Another group of studies approaches the problem by integrating the detected objects from different viewpoints. Franzel et al. [7] use X-Ray images of the same scene from different viewpoints and integrate them with a voting-based approach to find the object pose. Roig et al. [17] detect cars, buses and people by combining different detections from six cameras by using conditional random fields. Another approach was introduced by Viksten et al. [24], to detect objects from different views and integrate the information using a mean-shift clustering algorithm. Even if there are false detections in single views, detection is improved by integration. However, it is not mentioned how to overcome the cases where there is not a correct pose estimate in some of the views, which can occur due to the absence of the object in one of the views.

There have also been studies that used appearance-based models in multiple-camera setups. For example, Helmer et al. [9] combine different viewpoints by using the projections of the objects into 3D. They argue that any appearance-based method can be used. Their method maximizes the conditional likelihood of object detections. They do not use RGB-D cameras. In another study, Coates and Ng [4] use corresponding appearance features to compute the posterior pose probability. They use a pant-tilt-zoom camera and only one object category for experiments. Finally, Susanto et al. [20] combine the final pose estimate from each individual viewpoint into a single 3D location. VFH descriptors [18] are computed in the reconstructed 3D scene and are integrated with the results from a DPM object detector, where DPM uses a discriminatively learned part model with a latent SVM model. They perform intensive experiments with 4 Kinects, which result in interference. Therefore, there is significant noise in the depth data. We compare our results with this method in Sect. 4.

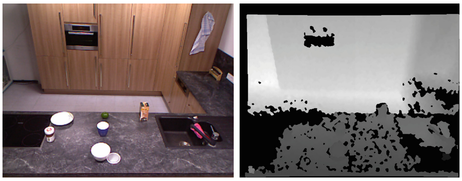


Fig. 2. The interference of multiple RGB-D cameras results in noisy depth data as seen on right. Gray areas have a valid depth, while the black regions do not provide any information about depth.

As mentioned previously, pose estimates of the objects are necessary from multiple views, and any pose estimation method can be used. There are some alternatives that can be used to detect textureless objects. For example, Papazov and Burschka [16] use an efficient RANSAC-like sampling strategy to establish correspondence between the scene and the model. However, this work requires a robust local descriptor like SHOT [23]. It can be difficult to find correspondences for object features without distinctive depth features. Furthermore, when multiple Kinects are used, 3D data may be noisy due to interference problems. Brachman et al. [2] use a single decision forest and use the minimization of an energy function which uses depth as one of the components. Background RGB-D images are necessary to train the objects. Although they use a uniform noise or a simulated plane, when the background has similar texture with the object, it may be difficult to find the object in such a setting. Also the simulated plane will be affected by interference problems during testing. It should be mentioned that although some studies obtain features by using learning algorithms like Convolutional Neural Networks [25], we prefer to use manually designed features. In another study, Tejani et al. [21] use LineMOD features [10] and adopt Latent-Class Hough Forests [8]. LineMOD matches viewpoint samplings of the object by using selected features. In LineMOD, if the surfaces of the objects don't have distinctive features, it can be difficult to detect objects. Another alternative is a probabilistic appearance model which is reported to estimate the poses of objects without texture [22]; however, it is not possible to combine multiple features if one of the attributes has noise, or unavailable. We introduce depth, color and surface normal attributes together with edge orientations into this method, details of which are explained in Sect. 2.2.

2 Probabilistic Appearance Based Estimation

In this section, we will first briefly explain the probabilistic model of appearance and present how we combine different features. Next, we will show the feature types that we used in this study for pose estimation from single views.

2.1 Probabilistic Model of Appearance

We assume that we want to find the pose of a previously-learned object in a given test scene. Let the features of the test scene t be denoted by $x_t^f = \{a_{xt}^f, p_{xt}\}$ where f is the feature type, a_{xt}^f is the appearance attribute, and $p_{xt} \in \mathbb{R}^+$ is the position of the feature in the image plane. A similar notation can be used for the features of the learned object l, $x_{lv}^f = \{a_{xlv}^f, p_{xlv}\}$, where v denotes the viewpoint of the learned object. Viewpoint is important because we are using the appearance of the object for detection. The viewpoint v includes the azimuth (θ), elevation (γ) and image-plane rotation (α) angles, and the distance d of the object to the camera (Fig. 3 left). The object is learned from multiple viewpoints at a known distance to the camera during training. The pose of the object can be found in 6DoF if the viewpoint v and the camera parameters are known and the object is at a known position p_{xt} in the test image.

We turn the set of image features into a distribution of features using the approach explained by Teney and Piater [22]:

$$\phi_t^f(x_t^f) = \int_{\mathcal{I}_t} \mathcal{N}(p_{x_t}^f, p_y^f, \sigma^f) \mathrm{K}^f(a_{x_t}^f, a_y) \, \mathrm{d}y \tag{1}$$

Here, $\mathcal{I}_t$ denotes the test image, and K is a kernel associated with the feature type f. Then, the distribution of training features $\phi_{lv}^f(x_l^f)$ can be obtained similarly. The similarity between the test scene and the learned object at viewpoint v is given as the cross-correlation between two distributions:

$$\left(\phi_t^f \star \phi_{lv}^f\right)(x_t) = \int_{\mathcal{I}} \phi_t^f(x_t + y) \phi_{lv}^f(y) \, \mathrm{d}y \tag{2}$$

As suggested by Teney and Piater [22], we use Monte Carlo integration for efficiency, which involves drawing samples y_i from $\mathcal{I}$. We obtain the cross-correlation of distributions for viewpoint v at image position x_t for feature type f as

$$\varPhi_{x_t,v}^f \approx \frac{1}{N_L} \sum_{y_i}^{y_L} \phi_t^f(x_t + y_i) \phi_{lv}^f(y_i), \tag{3}$$

where N_L is the total number of samples drawn from the image features. We combine different features using

$$\varPhi_{x_t,v} = \prod_f^F \varPhi_{x_t,v}^f (1 - \lambda^f) + \lambda^f, \tag{4}$$

where each type of feature is denoted by $f = 1, \ldots, F$, and λ^f is the parameter related to the probability of the absence of a feature. This parameter increases the possibility that the corresponding location will be considered as a candidate pose estimate even if there is no attribute a_x that supports the existence of a candidate pose at position x_t. The local maxima of $\varPhi_{x_t,v}$, which can be isolated by non-maximum suppression, constitute the pose estimates for the object. Each pose estimate is denoted by an ordered pair (x_t, v), and can be converted into a 6DoF pose via the camera parameters. The 6DoF pose estimates are denoted by $\mathbf{x}_e$ with a corresponding confidence score s_e. The score is the value of $\Phi_{x_t,v}$ at the local maxima points, which is the similarity between the estimated pose and the corresponding learned object. We make an assumption such that the similarity score is related to the confidence of the spatial pose of the object. When the score is high for a pose x_e, the confidence is also high.

2.2 Feature Types

In this section, we show the feature types that we used to recognize the objects. Note that the feature types can be extended for other studies in a straightforward fashion. We select the features which can be detected in textureless objects. For each feature type, a dedicated kernel $\mathrm{K}^f(a^f_{x_1}, a^f_{x_2})$ is used. All features are associated with a position $p_x \in \mathbb{R}^2$ in the image plane. An overview of the process can be seen in Fig. 3.

Edge Orientation. We use an intensity-based Canny edge detector [3]. Each edge point feature has an appearance attribute $a^\circ_x \in S^+_1$ giving the local orientation of the edge at a given position. The kernel uses a von Mises distribution on the half circle, which is defined as $K^\circ(a^\circ_{x1}, a^\circ_{x2}) = C_o \mathrm{e}^{\kappa_o \cos(a^\circ_{x1} - a^\circ_{x2})}$. Our distance measure can be said to be a general form of the directed chamfer distance [13]. C_o is a normalization constant.

Depth. Depth values are obtained from depth images. Each depth feature has only one depth value as an appearance attribute $a^d \in \mathbb{R}^+$. The kernel can be defined as $K^d(a^d_{x1}, a^d_{x2}) = C_d \mathrm{e}^{-(a^d_{x1} - a^d_{x2})^2}$. C_d is a normalization constant.

Color. The color feature $a^h \in [0, 1]$ is given by the hue component of the HSV color space. The kernel can be defined as $K^h(a^h_{x1}, a^h_{x2}) = C_h \mathrm{e}^{\kappa_h \cos(a^h_{x1} - a^h_{x2})}$. C_h is a normalization constant.

Surface Normal. The surface normals $a^n \in S^+_2$ are normal vectors at a point $\mathbf{p}$. The kernel can be defined as $K^n(a^n_{x1}, a^n_{x2}) = C_n \mathrm{e}^{\kappa_n \cos(\|a^n_{x1} - a^n_{x2}\|)}$. C_n is a normalization constant.

3 Multiple-View Integration

In this section, we explain how to integrate pose estimates from multiple views to obtain the actual pose of the object in 6D. We are going to use the pose estimates and the associated scores obtained with the approach explained in the previous section; however, it should be noted that any pose estimation method can be used. If there are no scores associated with the pose estimations, then we can assume a uniform probability distribution among all pose estimates.

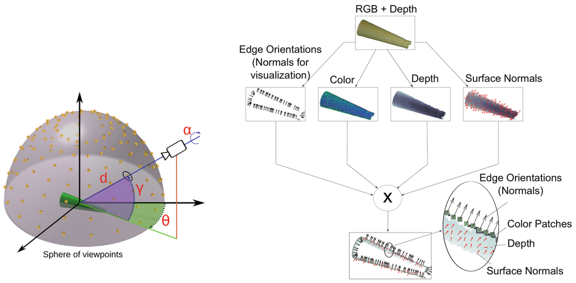


Fig. 3. Left: The object is at the center of the sphere. The dots on the sphere illustrate the viewpoints. Right: The feature types related to the learning of the object.

An overview of the integration process can be seen in Fig. 4. First, pose estimations are made for each view by using information obtained from sensors 1 and 2, only two of which are correct for each view. Each pose estimate from each view is integrated to obtain the integrated pose estimate surfaces. The highest values are the scores of final pose estimates, which are the first two diagonal elements in Fig. 4. If the correct estimation was made by only one of the views, it would still be possible for our proposed method to make a correct estimate, because a high score would dominate in the integrated surface. Now, we will explain this process in detail.

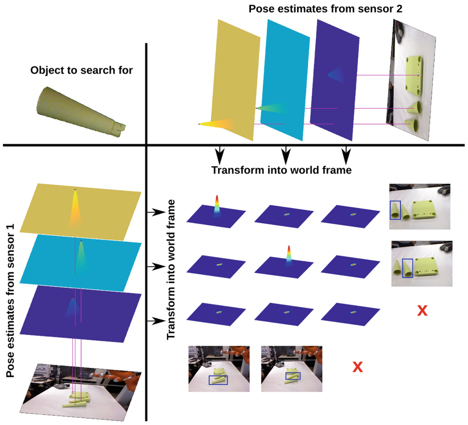

Pose estimates from sensor 2
Object to search for
Transform into world frame
Pose estimates from sensor 1
Transform into world frame
X
X

Fig. 4. Pose estimations are made for the object shown on the upper left. For example, there are three estimates for each view. The integration is made in the world frame. The process is shown in 2D for illustration purposes.

We have a set of pose estimates $\mathbf{x}_{vi}^{e}$ from view v_i, each associated with a score s_{vi}^{e}. During integration, we consider all the pose estimations from all the views, i.e. the target object can be seen and recognized correctly by any combination of the views. Therefore, first, we obtain all the subsets of the views, $V^p = \{(v^{p_1}, v^{p_2}, \ldots, v^{p_{N_p}}) : v^{p_j} \subseteq V, \forall j = 1, \ldots, 2^{N_v}\}$, where V is the set of all the views, $\|V\| = N_v$ is the number of views, v^{p_i} is one of the subsets, and $N_p = 2^{N_v}$ is the number of subsets. Next, we consider the set of all possible pose estimation combinations from view subsets V^p which can be defined as $C = \{(x^e_{v1}, \ldots, x^e_{vn}) : x^e_{vi} \in v_i, \forall v_i \in v^{p_j}, v^{p_j} \in V^p, \forall j = 1, \ldots, 2^{N_v}\}$. Each element $c_k \in C$ contains a set of pose estimates, which includes at most one estimate from each view. The total number of pose estimate combinations will be $\|C\| = \sum_{j}^{N_p} \prod_{v_i \in v^{p_j}} \|\mathbf{x}^e_{vi}\|$, where $\|\mathbf{x}^e_{vi}\|$ is the number of estimates for view v_i.

Now, we have a combination set of pose estimates. Next, we obtain a distribution in 6D for each pose estimate in v_i,

$$\Phi(x^e_{vi}) = \mathcal{N}(x^e_{vi}, \Sigma), \tag{5}$$

which is simply a Gaussian centered at the estimate in the ith view with a covariance of Σ. The covariance matrix is a 6×6 diagonal matrix and its diagonal values are selected to be equal to ${s^e_{vi}}^{-1}$.

After we obtain the distributions of each pose estimate, we use them to construct the distribution of the combined pose estimates $c_j \in C$,

$$\varphi(c_j) = \prod_{x^e_{vi} \in c_j} \Phi(x^e_{vi}). \tag{6}$$

The $\varphi(c_j)$ are what is visualized as surfaces in Fig. 4. We need to find the value which maximizes $\varphi(c_j)$ to obtain possible pose estimates for each c_j. This can be achieved by taking the derivative of $\varphi(c_j)$ with respect to $\mathbf{x}$:

$$\nabla\varphi(c_j) = \left[\frac{\partial\varphi(c_j)}{\partial x_1} \cdots \frac{\partial\varphi(c_j)}{\partial x_6}\right] = 0 \tag{7}$$

and solving it for each dimension:

$$\frac{\partial\varphi(c_j)}{\partial x_k} = \sum_{i=1}^{N_v} \frac{x_k - x^e_{k,vi}}{{s^e_{vi}}^{-1}} = 0 \tag{8}$$

For each pose estimate combination c_j we can find the $x^*_{c_j}$ that maximizes $\varphi(c_j)$ by solving this equation. The final pose estimate can be obtained by finding the maximum score among combinations C:

$$x^* = \arg\max_{x^*_{c_j}} \sqrt[\|v^{p_j}\|]{\varphi(c_j)\lambda_v} \tag{9}$$

Equation 9 ensures that pose estimations in a combination subset of views are not selected only because of the small number of views in the subset. $\|v^{p_j}\|$ is the number of views in the pose estimation combination subset and $\varphi(c_j) < 1$.

When an estimation is made by a combination subset c_m with large number of views, $\varphi(c_m)$ will be lower than an estimation made by a combination subset c_n with less number of views if the $\|v^{p_j}\|^{th}$ root of $\varphi(c_m)$ is not taken. $\lambda_v \in [0,1]$ is a parameter used to induce the estimations made with a smaller number of views. As λ_v gets closer to 0, combination subsets c_m with higher number of views are selected.

4 Experiments

In this section, the approach is evaluated in three different environments with different objects. In all the experiments, the necessary parameters $\sigma, \lambda_f, C_o, \kappa_o, C_d, C_h, \kappa_h, C_n, \kappa_n$ are obtained by cross-validation. The "probability of absence" parameter is set to $\lambda^f = 0.3$ for all features except edge orientations, where $\lambda^\circ = 0.0$. In Sect. 4.1, the probabilistic appearance pose estimation method is compared with other widely used pose estimation approaches. In Sect. 4.2, we mainly compare our method with another detection method. In Sect. 4.3, we give the results of the accuracy of pose estimation.

4.1 Single View Pose Estimation

In the first set of experiments, the poses of multiple objects are estimated in a cluttered scene [21]. There are 6 objects with multiple instances in each scene. The number of scenes are over 700 images for each object. The objects are learned by using the 3D object models as shown in Fig. 5. The training images are captured for azimuths in the range of $\theta \in [0, 2\pi]$ and elevations in the range of $\gamma \in [0, \pi/2]$ in 5-degree steps ($\delta\theta = 5°$, $\delta\gamma = 5°$). There are foreground occlusions, 2D and 3D clutters in the test scenes.

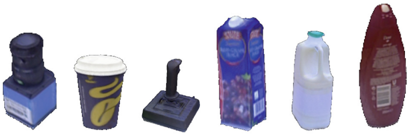


Fig. 5. Object models used in Sect. 4.1.

The comparison is made against two methods, LineMOD [10] and LCHF [21]. The measure defined in [11] is used to determine a successful pose estimation. For each object instance in each scene, there exists a ground truth rotation $\mathbf{R}$ and translation $\mathbf{T}$. If the estimated rotation and translation for the object with the model $\mathbf{M}$ are annotated as $\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}$ respectively, then the measure for pose estimation of symmetric objects can be given as

$$m = \operatorname*{avg}_{x \in \mathbf{M}} \left\| (\mathbf{R}x + \mathbf{T}) - \left(\hat{\mathbf{R}}x + \hat{\mathbf{T}} \right) \right\|, \tag{10}$$

and for non-symmetric objects as

$$m = \operatorname*{avg}_{x_1 \in \mathbf{M}} \min_{x_2 \in \mathbf{M}} \left\| (\mathbf{R}x_1 + \mathbf{T}) - \left(\hat{\mathbf{R}}x_2 + \hat{\mathbf{T}} \right) \right\|. \tag{11}$$

Estimation is a success if $m < k_m d$ where d is the diameter of the object and $k_m = 0.15$ in our comparison. The F1-scores for three methods can be seen in Table 1. As can be observed, three objects are recognized better with probabilistic appearance-based pose estimation method, while three objects are recognized better with LCHF. On average, accuracies are roughly equal. For PAPE, the best estimate performances are for the camera and coffecup with respect to LCHF. The superiority of PAPE for these objects can be related to the discriminative visual features. They have unique colors and their edges are visible under different viewing angles, which is important for edge orientations. On the other hand, the accuracy is lower especially for the milk bottle and juice carton. For the milk, the background has features similar the the milk, which inreases the rate of wrong estimates. For the juice carton, as can be seen in its model in Fig. 5, the visual features are not clear, which makes it difficult to discriminate its visual appereance features that are important for PAPE. It should also be noted that since multiple attributes are combined, the pose estimation accuracy in cluttered scenes increases with respect to the single-attribute method. As the PAPE results are comparable with other state-of-art methods for pose estimation, we can conclude that it can be used with the multi-view pose estimation method. Some of the results for the pose estimation in this set of experiments can be seen in Fig. 6.

4.2 Multi-view Detection

In the second set of experiments, we detect the location of the objects in different scenes using the MPII Multi-Kinect Dataset [20]. It is one of the few available datasets containing real RGB and depth images from different viewpoints for object recognition. The dataset contains 9 different object classes and a total of 33 scenes. Four Kinects are used to capture the scene, but only three of them

Table 1. F1-scores for Sect. 4.1

	PAPE	LCHF [21]	LineMOD [10]
Joystick	51.5	**53.4**	45.4
Camera	**80.7**	37.2	42.2
Coffee cup	**99.5**	87.7	81.9
Shampoo	**82.5**	75.9	62.5
Milk carton	27.2	**38.5**	17.6
Juice carton	41.2	**87.0**	49.4
Avg	**63.8**	63.3	49.8

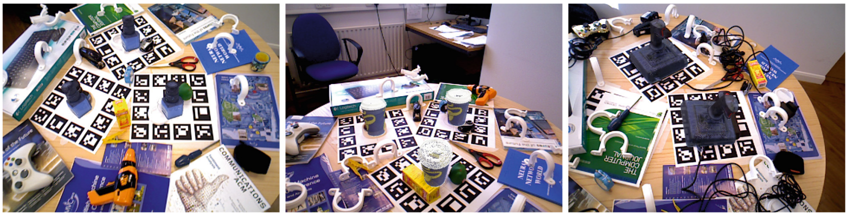


Fig. 6. Visualization of some of the results for Sect. 4.1. The estimates of the 3D object models are rendered on the image for visualization.

are used to recognize the objects, as one of them is used to obtain the ground-truth poses of the objects. The sensors interfere with each other; therefore, the quality of the depth data is poor. Another point to mention is that the provided calibration, which is obtained from the depth data, contains large errors reported as up to 13 cm in 3D space. The dataset includes two parts, for classification and detection respectively. In the classification part, there is only one object instance in the scene. We have used this part to learn the objects. We have used RGB-D images from three viewpoints ($N_v = 3$).

Since our method mainly finds the pose estimate of the target object, we have found the bounding box after finding the pose estimation for comparison reasons. We compared our results with those obtained by Susanto et al. [20]. We used RGB and depth images and calibration files provided with the dataset. The ground truth of the objects was also available with the dataset.

Susanto et al. [20] use a Deformable Parts Model (DPM) [6] together with VFH descriptors using reconstructed 3D scenes and estimate the poses by using the combinations of these features from multiple views (mDPM + mVFH). The comparison of the results can be seen in Table 2. The first column reproduce the average precision (AP) results from [20], and the remaining columns indicate the AP results obtained using our PAPE approach with different numbers of cameras. The AP is computed as described in [19] with a bounding-box overlap of 50 % of the detected object [5].

The APs are generally higher than the results obtained by Susanto et al. [20]. Avocado, bowl, plate, cup and sponge are detected with a high AP. An advantage of using multiple cameras is occlusion handling. For example, the bowl is detected successfully as shown in Fig. 7. It can also be observed that when the number of cameras increases, the accuracy of detections also increases. Therefore, we can suggest that the detection rate of the probabilistic, appearance-based pose estimate will increase if it is used with multiple cameras; however, it should be mentioned that the effect of adding new cameras on pose estimation performance reduces with the number of views. It can be reasoned that new views do not provide new information regarding the scene. Quantitatively, we can state that with our approach the accuracy increases by almost 24 % when multiple cameras are used instead of a single camera. One of the possible reasons is that the proposed method uses simple appearance-based attributes of the scene, so that

Table 2. Detection results and comparison (AP in %)

	mDPM + mVFH [20]	mPAPE 3 Cams	mPAPE 2 Cams	mPAPE 1 Cam
Avocado	**100.0**	**100.0**	99.2	79.7
Bowl	87.0	**99.7**	**99.7**	90.3
Coffee box	80.0	**92.4**	87.3	80.0
Coffee can	89.6	89.9	91.1	**98.1**
Cup	**100.0**	97.6	96.2	82.8
Nutella can	89.2	**93.6**	87.0	58.4
Plate	90.2	**98.1**	97.2	82.3
Spice can	**98.5**	96.8	97.4	78.0
Sponge	97.0	**97.4**	95.4	48.3
Mean	92.4	**96.2**	94.5	77.5

objects without texture can be recognized with a higher performance, while the method proposed by Susanto et al. [20] uses VFH, which would need more complex shape features. This may be one reason why mDPM + mVFH has better performance when estimating objects like the spice can and the coffee can, while mPAPE has a better recognition rate for textureless objects like plate and bowl. Surprisingly, for coffee cup, the single-view method performs better with the proposed approach. This may be due to false detections with high scores in the other views which resemble the appearance of the coffee can in the scene. Overall, it can be summarized that mPAPE can estimate the poses of the objects even if the information is partly absent/noisy for some of the features in the scene, e.g. depth features in the mentioned dataset.

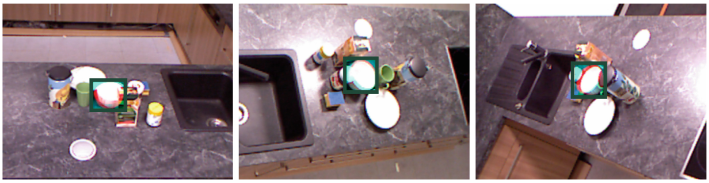


Fig. 7. Partially occluded object bowl can be detected successfully. The images are cropped for illustration purposes.

A common cause of failure is the resemblance of objects in terms of the visual features used in our approach. For example the cup and bowl can be mistaken for each other as shown in Fig. 8. Both of them have a convex inner surface and their inner surfaces are white which results in similar visual appearance. Other errors are due to object viewpoints that were not learned during training. There is only a limited number of viewpoints present in the dataset.

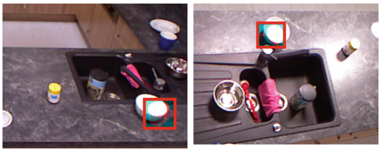
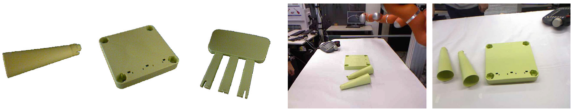


Fig. 8. A wrong detection. Bowl is detected as a cup.

4.3 Multi-view Pose Estimation

In the third set of experiments, we estimate the poses of textureless objects. IKEA chair parts are used as shown in Fig. 9a. The training images are captured for azimuths in the range of $\theta \in [0, 2\pi]$ and elevations in the range of $\gamma \in [0, \pi/2]$ in 5-degree steps ($\delta\theta = 5°$, $\delta\gamma = 5°$).



(a) Objects (b) Test scenes by two cameras.

Fig. 9. Experimental setup for the pose estimation experiments.

There are three different object types and six different object instances in the test scenes. Their poses are estimated in 13 different scenes. Since we used the KUKA Light-Weight-Robot arm for recording the poses of the objects, only those views that lie in the workspace of the robot were used for pose estimation. The poses of the objects are determined in the reference frame of the robot; therefore, the errors in the calibration of the camera position will also contribute to the error of the final evaluation. It should be noted that, in other studies, the reference frame is generally the camera itself. The scenes are captured using two Kinects as seen in Fig. 9b. To avoid interference issues, we used the freenect library, which has the capability of shutting down the IR light of the Kinects. However, due to delayed onset times of the IR light in the camera, some of the depth images do not contain sufficient information, as seen in Fig. 10.

In the first part of the evaluation, we compare the results obtained from single and multiple cameras. We used training data with coarse ($\delta\theta, \delta\gamma = 20°$) and dense ($\delta\theta, \delta\gamma = 5°$) angular spacing to see the capacity of our approach under both conditions. LineMOD [10] is also used to estimate the poses of the objects; however, since the bottom and back parts of the chair have no discriminative surface normal values or any discriminative visual feature from the surrounding, the detection rate performance was low with LineMOD; therefore, only the pose estimation for the chair leg is compared with the proposed method by using two cameras. The results are given in Table 3. As can be seen, the average error gets smaller with angular sampling step size. The error is smallest when the

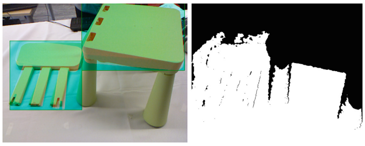


Fig. 10. Left: The pose estimates for two objects. Right: Corresponding depth image. Black regions contain no depth information.

Table 3. Average pose estimation errors

	Single cam	Multi-cam	LineMOD singleview	LineMOD multiview
Coarse	0.0226 m, 11.8°	0.0190 m, 11.1°	–	–
Dense	0.0186 m, 10.0°	0.0179 m, 9.3°	0.0400 m, 13.8°	0.0347 m, 12.6°

multi-view method is used. There is a decrease of around 10 % in orientation estimation error between dense single-camera and dense multi-camera settings. The pose estimation with LineMOD has a performance worse than mPAPE. This is probably due to the lack of discriminative surface features of the chair leg, which makes it harder for LineMOD to make precise pose estimates. The multi-view method increases the coarse pose estimation by approximately 15 % in position. An increase of this magnitude is not observed in dense estimation.

Using multiple-camera pose estimation with dense training data, we obtained the errors in translation and orientation shown in Fig. 11. For mPAPE, the errors are concentrated at 0.015 m, while for lineMOD, the errors are concentrated around 0.03 m. As it can be seen, mPAPE has a higher estimation accuracy; however, there are still errors higher than 0.01 m which can cause problems if a high precision estimation is necessary. Multiple reasons exists for these errors. The first is the difficulty of obtaining stable features for recognizing textureless, flat objects. Features can be different under changing illumination conditions and the noise in the depth data due to multiple RGB-D cameras. We have used a probabilistic, appearance-based pose estimation method to overcome this. The second source is related to occlusion of the objects in some of the scenes. Unsurprisingly, it has been observed that the error increases when the object is occluded in one of the views. The third reason is the calibration error of the cameras. The poses of the objects are recorded with respect to the robot frame. When the estimated pose is transformed into the robot frame, this affects the result.

In all sets of experiments, pose estimation integration from multiple cameras with our approach provided higher accuracy and precision with respect to single-camera pose estimation. The improvement in detection rate, around 24 %, is even higher than the improvement in the pose estimation, which is around 10 %.

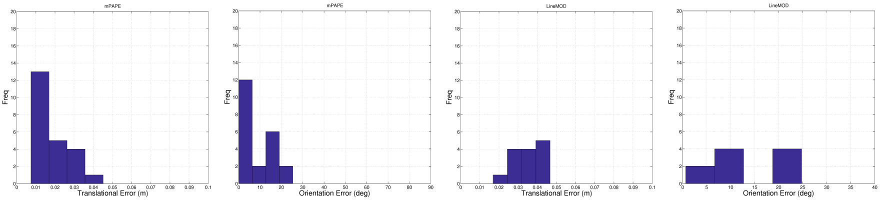


Fig. 11. Histogram of translational and orientation errors of mPAPE and lineMOD for chairleg parts respectively. Errors of up to 0.04 m in translation and 20° in orientation can be observed for mPAPE, while errors upto 0.05 m in translation and 25° in orientation can be observed for lineMOD method.

5 Conclusion

We proposed a method for integrating pose estimations from multiple sensors. A probabilistic appearance-based pose estimation method has been improved to combine multiple attributes of the scene, even if one of the features (e.g. depth information) is noisy or incomplete in the scene.

We have developed a method to integrate poses from multiple views and used it with PAPE; however, it should be noted it was also possible to use with other pose estimation methods (e.g. lineMOD [10] which had lower performance as shown in the experiments). The results show that mPAPE can achieve high accuracies when pose estimations are integrated with the approach proposed in this paper; and they are comparable to or exceed state-of-the-art results. Furthermore, errors in object pose estimation are reduced with multiple cameras.

Acknowledgement. The research leading to this work has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 610878, 3rd HAND.

References

1. Aldoma, A., Thomas, F., Vincze, M.: Automation of ground truth annotation for multi-view RGB-D object instance recognition datasets. In: IEEE International Conference on Intelligent Robots and Systems. pp. 5016–5023 (2014)
2. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6D object pose estimation using 3D object coordinates. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8690, pp. 536–551. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10605-2_35
3. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 679–698 (1986)
4. Coates, A., Ng, A.Y.: Multi-camera object detection for robotics. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 412–419 (2010)
5. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge-a retrospective. Int. J. Comput. Vis. **111**, 98–136 (2014)

6. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
7. Franzel, T., Schmidt, U., Roth, S.: Object detection in multi-view X-Ray images. In: Pinz, A., Pock, T., Bischof, H., Leberl, F. (eds.) DAGM/OAGM 2012. LNCS, vol. 7476, pp. 144–154. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32717-9_15
8. Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V.: Hough forests for object detection, tracking, and action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2188–2202 (2011)
9. Helmer, S., Meger, D., Muja, M., Little, J.J., Lowe, D.G.: Multiple viewpoint recognition and localization. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6492, pp. 464–477. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19315-6_36
10. Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 858–865. IEEE (2011)
11. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7724, pp. 548–562. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37331-2_42
12. Izadi, S., Davison, A., Fitzgibbon, A., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D.: Kinect fusion: real-time 3D reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST 2011, p. 559 (2011)
13. Liu, M.Y., Tuzel, O., Veeraraghavan, A., Chellappa, R.: Fast directional chamfer matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1696–1703. IEEE (2010)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
15. Mustafa, W., Pugeault, N., Kruger, N.: Multi-view object recognition using viewpoint invariant shape relations and appearance information. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4230–4237 (2013)
16. Papazov, C., Burschka, D.: An efficient RANSAC for 3D object recognition in noisy and occluded scenes. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6492, pp. 135–148. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19315-6_11
17. Roig, G., Boix, X., Shitrit, H.B., Fua, P.: Conditional random fields for multi-camera object detection. In: 2011 International Conference on Computer Vision, pp. 563–570, September 2011
18. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the viewpoint feature histogram. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2155–2162 (2010)
19. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill Inc., New York (1983)

20. Susanto, W., Rohrbach, M., Schiele, B.: 3D object detection with multiple kinects. In: Fusiello, A., Murino, V., Cucchiara, R. (eds.) ECCV 2012. LNCS, vol. 7584, pp. 93–102. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33868-7_10
21. Tejani, A., Tang, D., Kouskouridas, R., Kim, T.k.: Latent-class hough forests for 3D object detection and pose estimation. In: European Conference on Computer Vision, pp. 462–477 (2014)
22. Teney, D., Piater, J.: Multiview feature distributions for object detection and continuous pose estimation. Comput. Vis. Image Underst. **125**, 265–282 (2014). https://iis.uibk.ac.at/public/papers/Teney-2014-CVIU.pdf
23. Tombari, F., Salti, S., Stefano, L.: Unique signatures of histograms for local surface description. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6313, pp. 356–369. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15558-1_26
24. Vikstén, F., Söderberg, R., Nordberg, K., Perwass, C.: Increasing pose estimation performance using multi-cue integration. In: IEEE International Conference on Robotics and Automation, pp. 3760–3767 (2006)
25. Wohlhart, P., Lepetit, V.: Learning descriptors for object recognition and 3D pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3109–3118 (2015)
26. Yang, A., Maji, S., Christoudias, C., Darrell, T., Malik, J., Sastry, S.: Multiple-view object recognition in smart camera networks. In: Bhanu, B., Ravishankar, C.V., Roy-Chowdhury, A.K., Aghajan, H., Terzopoulos, D. (eds.) Distributed Video Sensor Networks, pp. 55–68. Springer, London (2011)