Pose Estimation using Local Structure-Specific Shape and Appearance Context

Anders Glent Buch[1], Dirk Kraft[1], Joni-Kristian Kamarainen[2], Henrik Gordon Petersen[1] and Norbert Krüger[1]

Abstract— We address the problem of estimating the alignment pose between two models using structure-specific local descriptors. Our descriptors are generated using a combination of 2D image data and 3D contextual shape data, resulting in a set of semi-local descriptors containing rich appearance and shape information for both edge and texture structures. This is achieved by defining feature space relations which describe the neighborhood of a descriptor. By quantitative evaluations, we show that our descriptors provide high discriminative power compared to state of the art approaches. In addition, we show how to utilize this for the estimation of the alignment pose between two point sets. We present experiments both in controlled and real-life scenarios to validate our approach.

I. INTRODUCTION

The problem of determining the alignment transformation between two model surfaces has undergone extensive research in the computer vision community. In robotic manipulation applications, the instantiation of an object model into a scene is crucial to the success of further processing tasks. A relevant application is grasping and manipulation of objects, which requires a great deal of accuracy from the vision system. Accuracy in this process is also crucial for more high-level manipulation tasks, such as placing or assembly operations.

In the registration or stitching problem, multiple views of the same object or scene model are used for building a more complete representation. This requires very accurate alignments between the views in order for the result to be usable. The same method can be applied for the estimation of the external camera parameters in a multi-camera setup.

Methods for solving these problems have been applied both in the 2D image domain as well as in 3D using range data or RGB-D data where color information is also available. For these domains, a feature-driven approach is commonly taken, which is based on an attempt to remove false feature correspondences between the two models, thereby making it possible to compute the alignment transformation using the remaining true point to point correspondences. It is widely accepted that local descriptors provide the highest stability in this process due to their tolerance towards clutter and occlusions [1].

For image data, interest points or keypoints are often being detected and are afterwards augmented with contextual image information using the appearance of the local

[1]Cognitive Vision Lab, University of Southern Denmark, 5230 Odense, Denmark. {anbu,kraft,hgp,norbert}@mmmi.sdu.dk
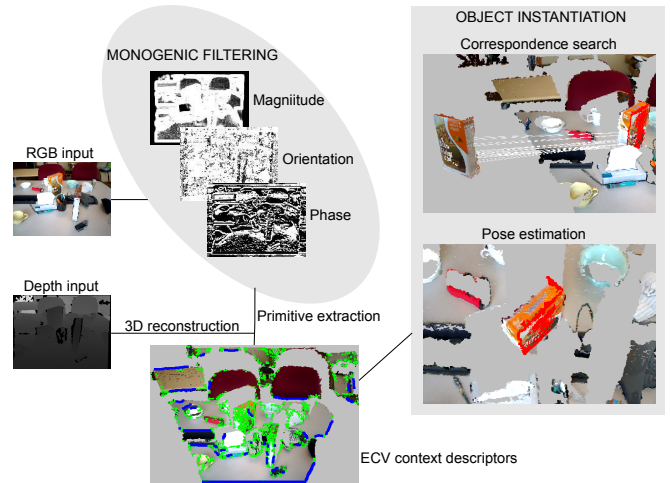[2]Tampere University of Technology, 33720 Tampere, Finland. joni.kamarainen@tut.fi

Fig. 1. Processing pipeline of our pose estimation system. Left: input RGB-D images. Top middle: split of identity of appearance information by monogenic filtering, resulting in an image triple of local magnitude, orientation and phase. Bottom middle: ECV primitive extraction and their 3D reconstructed counterparts. Line segments are marked by blue, and texlets by green. Using this data, ECV context descriptors are generated. Top right: ECV context descriptor correspondences between a stored object model and the input scene. Bottom right: alignment pose estimation between the object and the scene.

neighborhood for generating the final image descriptor [2]–[6]. The keypoint detection is done to locate distinct interest points, thus making the extraction process approximately viewpoint-invariant. Discriminative power is ensured by building a description of the appearance around the keypoint. Keypoint-based model descriptions tend to be quite sparse, since a higher density would decrease distinctiveness of the individual feature descriptor.

For 3D data or point clouds, a variety of shape descriptors have been developed over the last couple of decades [7]–[12]. These are often built for the complete data set, i.e. at every point on the model, although feature selection methods exist [13], [14]. For shape descriptors, geometric invariants such as relative distances or angles can be used to describe the local neighborhood of a point. Since shape descriptors are often created for all points, these representations are dense.

In this paper, we introduce a new variant of descriptors for 3D models containing RGB data. The input to our feature processing can be either a dense stereo reconstruction or an RGB-D image (see Fig. 1), providing both appearance and shape data. Our aim is to combine inputs from both the appearance and the shape domain in an efficient manner.

As will be detailed in Sect. II, keypoint detection is done

by classifying individual pixels into different categories; this is achieved by the use of an Early Cognitive Vision (ECV) system. The density of an ECV representation falls between sparse image keypoint descriptors and dense 3D shape descriptors. This has the advantage that the model shape is captured in greater detail while maintaining a high level of discriminative power. To further increase the latter, we define context descriptors on top of the ECV model. As will be shown in the following sections, all this together allows for efficient pose estimation, both in terms of accuracy and speed.

Earlier works have applied ECV features for a range of computer vision applications. For the purpose of accurate scene representation, [15] presents a real-time extraction method for ECV texture primitives, providing uncertainty information for each primitive. In [16], egomotion estimation is the target, and ECV edge structures are used for representing edge information in the observed scene. In [17], ECV primitives are embedded in a probabilistic framework which allows for object recognition using generative models. Recently, ECV features have been applied for the task of grasping unknown objects [18], where a scene representation based on both edge and texture information is used for directly generating stable grasps.

The contribution of this paper is a model description containing both appearance and shape information and its applicability for different pose estimation tasks, such as object instantiation, scene registration and camera calibration as described in Sect. IV. We present an image processing pipeline which generates features both in the edge and in the texture domain. Using these features, we build a novel type of local 3D descriptor, utilizing the complementary power of both appearance and shape information. The result is a description with high discriminative power. We use our representation in a speed-optimized RANSAC [19] procedure, which shows the practical usability of our system. We argue for the efficiency of our representation in three ways: 1) we use both appearance and shape for describing a point, 2) keypoints are classified into edge/texture types, providing a structure-dependent descriptor, and 3) the keypoint density is high, allowing for more shape information than many other image descriptors.

We perform validation of our system both in a controlled experimental setup as well as for real-life scenarios. We compare against state of the art approaches and show that our representation can deal with a range of estimation problems more efficiently than other representations. We support this claim by showing 1) that our descriptors provide a high number of true correspondences under large viewpoint changes, and 2) how to efficiently solve the pose estimation problem between two models with large observation discrepancies.

The paper is structured as follows: Sect. II describes the acquisition of ECV context descriptors in a formal manner. In Sect. III, we motivate the use of our representation for efficient pose estimation. Experimental results are presented in Sect. IV, and conclusions are drawn in Sect. V.

II. Feature descriptors

The work presented here builds on top of the observation space contained in an ECV model. We start by shortly describing the ECV feature extraction process and then move on to a presentation of our context descriptors.

A. ECV primitives

The atomic features created by the ECV process are referred to as *primitives*. ECV primitives are extracted from an image by a rotation-invariant filtering method called the monogenic signal [20] and later reconstructed in 3D. The monogenic signal filtering extracts a triplet of local magnitude, orientation and phase (MOP) at each pixel location, based on the spectrum of the appearance of the local neighborhood. This expansion of dimensionality is referred to as a *split of identity*.

The MOP image is subdivided using a hexagonal grid, and keypoints are localized as the pixel with maximum magnitude response in each grid cell. The MOP splitting of an image motivates the concept of *intrinsic dimensionality* (ID) [21]. The measure of spectral variance in a cell gives information about the dimensionality of the local image structure. A small variation in magnitude indicates a homogeneous patch, whereas a small variation in orientation indicates an edge or a line structure. Textured areas are characterized by large variations in both magnitude and orientation. By appropriately thresholding the ID values in a cell, the cell can be classified as belonging to either a homogeneous patch, an edge or a texture region [22]. This gives low-level, but valuable information about what sensing modality different parts of the image represent. Since pixels belonging to homogeneous patches are inherently ambiguous, only edge regions and texture regions are used in this work.

Fig. 2 shows a visualization of extracted edge/texture primitives of an example object. We refer to edge primitives as *line segments*, or simply *segments* (blue), and textured surface primitives as *texlets* (green). In both this figure and in Fig. 1, the 3D reconstruction process is done directly using the shape data from an aligned depth sensor. Recently, the extraction of 3D ECV primitives from RGB-D data has been implemented on GPU, allowing for real-time operation [15].

Fig. 2. Visualization of different ECV features. Left: Input RGB-D model. Middle: ECV segments in blue. Right: ECV texlets in green.

Each ECV primitive includes a geometry and an appearance component. At this point, the geometry part consists of the 3D position and orientation. For segments, orientation is defined as the direction vector along the edge, and for texlets it is defined as the normal vector of the local surface region.

B. ECV context descriptors

In order to increase the discriminative power of ECV features, we generate an appearance- and geometry-based description of the spatial neighborhood of a feature. At each feature location, we calculate relations between all feature point pairs on the ECV model inside a Euclidean neighborhood of radius r and bin the results into histograms. We use the following geometric relations between two points $\mathbf{p}_1$ and $\mathbf{p}_2$ with orientations $\mathbf{o}_1$, $\mathbf{o}_2$:

- Cosine between orientation vectors: $R_{G1} = \mathbf{o}_1 \cdot \mathbf{o}_2$.
- Cosine between first orientation vector and point to point direction vector: $R_{G2} = \mathbf{o}_1 \cdot \frac{(\mathbf{p}_2 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1)\|}$.
- Cosine between second orientation vector and point to point direction vector: $R_{G3} = \mathbf{o}_2 \cdot \frac{(\mathbf{p}_2 - \mathbf{p}_1)}{\|(\mathbf{p}_2 - \mathbf{p}_1)\|}$.

Note that we use the general term *orientation* vector, since this can be either a surface normal (texlets) or a direction (segments). In Fig. 3, these geometric relations are visualized for a pair of texlets. An important detail is the ordering of any two points. The geometric relations R_G are asymmetric in the sense that their sign changes depending on which feature point is regarded as the first ($\mathbf{p}_1$) and the second ($\mathbf{p}_2$). We resolve this by selecting $\mathbf{p}_1$ as the point closest to the source feature for which we are calculating the context descriptor.
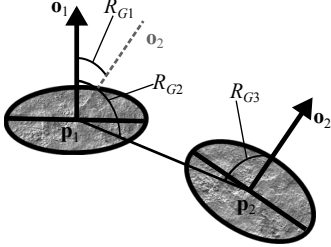

Fig. 3. Three geometric relations used between feature points, in this case texlets with an associated normal vector. Note that we do not use the value of the angle, but instead the cosine for saving computation time.

For the appearance part, we create individual histograms for all three RGB color channels. As for the geometric relations, we take all possible point pairs in the region and calculate the three intensity gradients. For the three color channels, we denote these appearance relations as R_{A1}, R_{A2} and R_{A3}. Again, we order points in pairs according to the distance to the source point. All in all, this gives six separate histograms, three geometry-based R_G and three appearance-based R_A, which are all assembled in the final descriptor vector. Each histogram consists of 16 bins, resulting in an overall descriptor dimension of 96. We use $r = 0.025$ m throughout the rest of this paper, which we have found to be a reasonable compromise between locality and discriminative power.

We stress that in contrast with many other works on 3D shape description, our context descriptor formulation operates only on the ECV feature points (segments and texlets), and not on the underlying point cloud. This has two advantages: 1) the number of points in the neighborhood is reduced, leading to a computational speedup, and 2) by using points that are classified into line/texture structures, we avoid the use of homogeneous surface points, which do not add to the discriminative power. The latter statement is justified in Sect. IV-A. We have deliberately kept the context descriptor formulation relatively simple, both to speed up the process, but also for showing the potential of ECV features for reliable context description. Clearly, this can be improved further by a systematic evaluation of alternative operators for describing feature contexts.

The extraction of a complete ECV context representation for a typical Kinect scene takes approximately 1 s, and we are currently working on porting the implementation to GPU for real-time extraction.

III. Pose estimation

In this section, we outline a simple, robust estimation algorithm for solving the pose estimation problem. Formally, the goal is to estimate a transformation $\hat{\mathbf{T}} \in SE(3)$ that minimizes the sum of squared distances between each point $\mathbf{p}$ on an object model $\mathbf{P}$ and the corresponding point $\mathbf{q}$ in the scene model $\mathbf{Q}$:

$$\hat{\mathbf{T}} = \underset{\mathbf{T}}{\operatorname{argmin}} \; \varepsilon(\mathbf{T}) = \underset{\mathbf{T}}{\operatorname{argmin}} \sum_{\mathbf{p} \in \mathbf{P}} (\mathbf{Tp} - \mathbf{q})^2 \qquad (1)$$

In the above equation we use the homogeneous representation of points to allow for the matrix-vector multiplication. The problem stated above is often addressed using a robust and outlier-tolerant method, such as RANSAC. A common way of treating this issue is based on feature correspondences, where the following is run iteratively:

1) Find $n \geq 3$ random object points in $\mathbf{P}$ and their corresponding points in $\mathbf{Q}$ by nearest neighbor matching of $SE(3)$-invariant feature descriptors.
2) Estimate a hypothesis transformation $\hat{\mathbf{T}}$ using the n sampled correspondences.
3) Apply the hypothesis transformation to the object model $\mathbf{P}$.
4) Find inlier points by spatial nearest neighbor search between the transformed object and the scene $\mathbf{Q}$, followed by Euclidean thresholding. If the number of inliers is too low, go back to step 1.
5) Re-estimate a hypothesis transformation using the inlier point correspondences.
6) Measure $\varepsilon(\hat{\mathbf{T}})$ using the inliers, and if it attains the smallest value so far, set $\hat{\mathbf{T}}$ as the resulting transformation.

In many cases, the algorithm can be optimized by stopping if ε falls below a predefined convergence threshold. Otherwise, the algorithm runs for the specified number of iterations.

We apply a modification to the RANSAC pose estimation procedure above by enforcing a low-level geometric constraint after the first step of each iteration. To achieve this, we exploit the fact that distances are preserved under transformations by isometric elements of $SE(3)$. Specifically, we make a simple check of the ratio between the edge lengths of the virtual polygons formed by the n sampled points on

both the object and the scene models. We denote the points sampled by feature correspondences as $\mathbf{p}_i, \mathbf{q}_i$, $i \in \{1, \ldots, n\}$. The edge lengths on the object polygon are given as $d_{\mathbf{p},i} = \|\mathbf{p}_{i+1 \bmod n} - \mathbf{p}_i\|$, likewise for the scene polygon edge lengths $d_{\mathbf{q},i}$. We then calculate the relative dissimilarity vector δ by ratios between the n polygon edge lengths:

$$\delta = \left[\frac{|d_{\mathbf{p},1} - d_{\mathbf{q},1}|}{\max\left(d_{\mathbf{p},1}, d_{\mathbf{q},1}\right)} \quad \cdots \quad \frac{|d_{\mathbf{p},n} - d_{\mathbf{q},n}|}{\max\left(d_{\mathbf{p},n}, d_{\mathbf{q},n}\right)} \right] \quad (2)$$

In case of a perfect match between two polygons, δ is identically zero. In practice, we can expect the largest deviation to be below a certain threshold t_{poly}:

$$\|\delta\|_{\infty} \leq t_{poly} \quad (3)$$

Our modification is therefore to insert the following step between step 1 and 2:

- Calculate the dissimilarity vector δ between the sampled polygon edge lengths. If $\|\delta\|_{\infty} > t_{poly}$, go back to step 1.

This verification step is significantly cheaper than enforcing the full geometric constraint using steps 2-6. Under the assumption that this step does not filter out hypothesis poses that align the models correctly, we can expect the same probability of success, but in much shorter time. Clearly, this assumption only holds as long as we can expect fairly accurate geometric observations such that the polygons are indeed isometric. If the sensor used exhibits a large depth error or produces a large distortion effect, the threshold t_{poly} would need to be set higher to allow for these inaccuracies. With the quality of sensors available today, we expect this problem to be of limited extent. We use $t_{poly} = 0.25$, thereby allowing for a maximal edge length dissimilarity of 25 %.

A large array of modifications to the original RANSAC has been proposed. These are based on different strategies, e.g. preemption [23], local optimization [24] and progressive sampling [25], only to name a few. Our modification is based on the simple criterion that wrong hypotheses should be filtered out immediately, thus saving time for generating a higher number of hypothesis poses. Our approach can be regarded as being hierarchical in the rejection phase in the sense that we introduce a preliminary low-level polygon-based rejection, which does not require a pose, before the usual inlier-based rejection phase.

The formula for calculating the required number of RANSAC iterations k given a desired success probability p and an expected inlier fraction w is [19]:

$$k = \frac{\log\left(1-p\right)}{\log\left(1-w^n\right)} \quad (4)$$

In our experiments, we sample $n = 3$ points in step 1. We use a conservative inlier fraction estimate of $w = 0.05$ and a desired success rate of $p = 0.99$, giving $k \approx 37000$. We set the Euclidean inlier threshold to 0.01 m, and the required number of inlier points is set to 50 % of the total number of object model points.

IV. EXPERIMENTAL RESULTS

We present three different experimental validations of our approach. In Sect. IV-A, we present a systematic evaluation method for determining the amount of true feature correspondences between two models. In Sect. IV-B, we apply our estimation procedure to the problem of registering two views of a scene. Finally, we show in Sect. IV-C results from our own setup. All experiments have been performed on a PC equipped with an 2.2 GHz Intel i7 processor with four cores (i7-2720QM). For speeding up nearest neighbor searches, we use the Fast Library for Approximate Nearest Neighbors [26].

A. Discriminative analysis

A key element of efficient pose estimation is the establishment of correspondences between the models to be aligned. In this initial experiment, we outline an algorithm for testing correspondences between two models. In general, we cannot a priori determine whether two points correspond; indeed, this is the goal of the estimation process. In [27], a correspondence is established during estimation if the ratio between the closest and the second closest feature matching distance is low. This ensures uniqueness of a correspondence, but only increases the probability of a correspondence actually being true.

Here we reverse the process: we start by performing alignment of the two models, and afterwards we check which of the originally calculated feature correspondences were correct, simply by thresholding the Euclidean distance between point pairs matched in the initial correspondence calculation phase. This is inspired by the correspondence rejection implemented in the Point Cloud Library (PCL) [28]. More specifically, we do the following:

1) Generate feature descriptors for both models and calculate the nearest matching scene feature of each object feature. This is the initial, hypothesized correspondence set C_{hyp}.
2) Perform accurate alignment of the two models such that each object surface point is well aligned to its true corresponding scene surface point (see text below).
3) Loop over all hypothesized correspondences and perform thresholding: if the Euclidean distance between the aligned matched point pair is below a given threshold, store this as a true correspondence. The result is a reduced set of correspondences C_{true}, likely to be true correspondences.
4) Calculate the *true correspondence* score as C_{true}/C_{hyp}.

If the models are sufficiently close to begin with, a local method such as Iterative Closest Point (ICP) [29], [30] can be used for step 2. Otherwise, a robust estimation method such as RANSAC can be applied, based on the correspondences generated in step 1. The Euclidean threshold in step 3 should be set to at least the expected mean accuracy of the alignment in order to compensate for estimation inaccuracies.

In Fig. 5, we show the whole process of alignment-based correspondence rejection between two views of the same
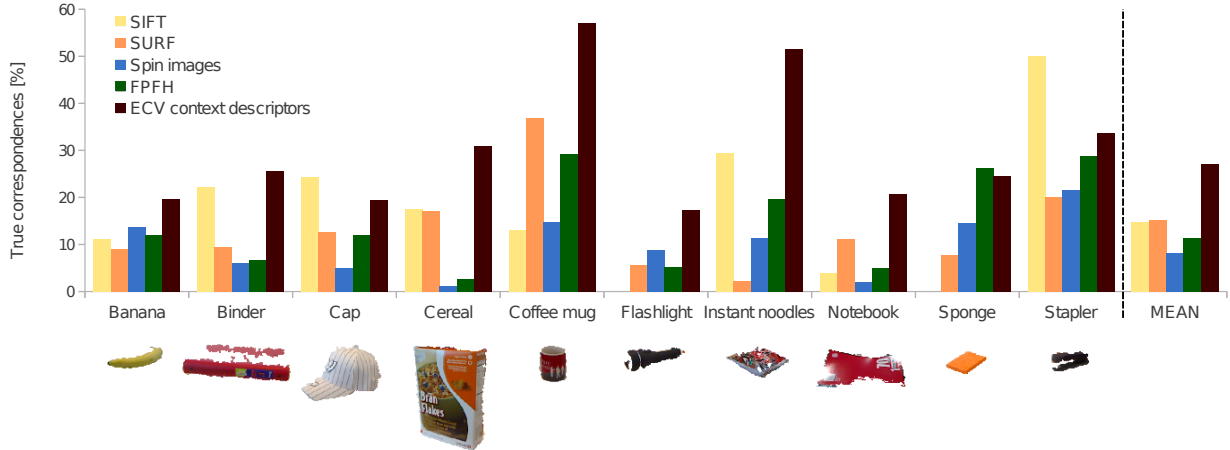
Fig. 4. Fraction of true correspondences using five different descriptors, including our own, for ten exemplary objects. The rightmost part of the chart shows the mean over all test cases.

object. The top figure shows a subset of the initial hypothesis set C_{hyp} by white lines, and the bottom figure a subset of the remaining correspondences C_{true}, again by white lines, as well as the ICP result by overlaying the aligned model in red.
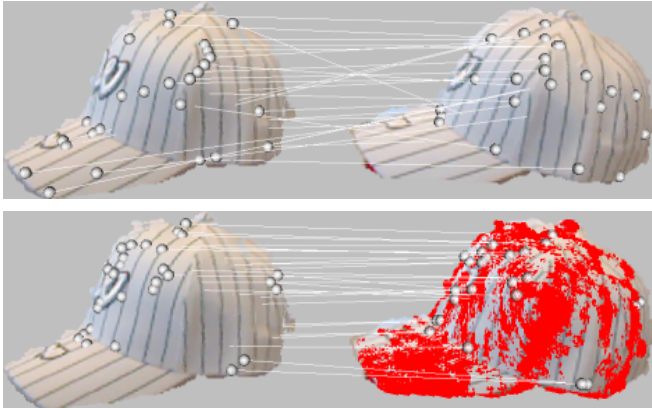

Fig. 5. Top: ECV context descriptor correspondences between the two object models. Bottom: filtered correspondences between the two object models after ICP alignment (aligned leftmost model overlaid in red) and Euclidean distance thresholding. For displaying purposes, only 25 randomly selected correspondences are shown in both cases.

The objects in both this figure and in Figs. 1 and 2 are taken from the RGB-D database [31] containing Kinect views of a large number of object models as well as some example scenes. All objects in the database are captured on a turntable with only a few degrees displacement between frames, and from three different elevations, giving several hundred views per object. Here we used the 1. and the 15. frame, which are displaced by approximately $30°$.

We have carried out this procedure for a set of randomly chosen objects from the RGB-D database, all using two views separated by 15 frames. For each object, we calculate the correspondence score using a Euclidean threshold of 0.01 m. We compare our ECV feature descriptors against state of the art descriptors in the image domain (SIFT [2] and SURF [6]) as well as in the 3D shape domain (Spin images [7] and FPFH [10]). For SIFT/SURF, we use OpenCV with standard settings. SURF does require a user-specified Hessian threshold for the keypoint detection, which we set to 500. For the shape descriptors, we use the PCL implementations with the radius set to $r = 0.025$ m. The calculated correspondence scores are reported in Fig. 4, with the mean over all scores shown in the rightmost part.

The results in Fig. 4 are not surprising. Indeed, shape-dominant objects such as "Banana" and "Sponge" are best described using shape descriptors. On the other hand, texture-rich objects such as "Binder" and "Cereal" are best captured by the appearance-based image descriptors. In one instance, "Sponge", the FPFH shape descriptor provided slightly better discriminative abilities than our proposed descriptor. This is expected since the appearance part of this object is highly ambiguous. For the objects "Cap" and "Stapler", SIFT produced better results. In the latter case, however, SIFT descriptors are calculated at only eight detected keypoints, making the object model extremely sparse. In comparison, ECV processing produces 113 keypoints for this object. We conclude from this study that our descriptors outperform the others in general, meaning on average over a range of objects with different shapes and appearances. The mean correspondence score from the total set is 27 %, whereas the next best, SURF, has a mean score of 15 %.

B. Scene registration and calibration

Algorithms for registering or stitching multiple images of a scene have been successfully implemented using robust 2D descriptors [32], [33]. The registration of multiple 3D models also has its practical usage, e.g. for building a model from several views. Another application is the calibration of a multi-camera setup, where multiple views of a scene must be registered in order to estimate the relative camera poses.

If enough shape information is available in the scene, the relative camera transformation can be determined very

accurately without the use of markers. In Fig. 6, we show the result of applying our modified RANSAC to two views of a scene with a wide baseline separation. For these kind of tasks, we lower the required number of inliers due to the limited overlap. In this experiment, we set the inlier fraction to 10 % of the number of points in the leftmost scene.

Fig. 6. Registration of two different views of the same scene using our optimized RANSAC algorithm. Top: input Kinect views, showing 25 randomly selected correspondences computed using ECV context descriptors. Bottom left: registration result with the aligned version of the leftmost scene view in the top image overlaid in red. Bottom right: zoom of the aligned point sets. Note the high quality of the alignment, which is performed without refinement.

We stress that we do not perform refinement of the results reported here, yet due to the high number of iterations, we achieve very accurate alignments. The registration time is below 1 s, which is achieved partly by our modified RANSAC procedure, partly by the low density of the ECV representation. The leftmost and rightmost scene in Fig. 6 are described by 1255 and 1448 ECV context descriptors, respectively.

In Tab. I, we show two relevant statistics of our modified RANSAC compared to the standard RANSAC algorithm. The numbers are the means of 100 runs of the above estimation problem. To be able to compare timings, we set the iteration count to 5000 for both cases. We observe that on average, our RANSAC performs almost 15 times faster than standard RANSAC. This is achieved without compromising the quality of the alignment, as can be seen by the mean fit error, which is calculated as the mean ε over the number of thresholded inliers.

C. Experiments from real setup

In this section, we present object pose estimation results from a robotic setup in our own laboratory. For all the experiments, we have conducted, estimation time for a single object has remained below 0.1 s. We currently use the pose estimation algorithm presented in this work for benchmarking robotic grasping strategies developed at our institute

TABLE I

TIMINGS AND MEAN INLIER FIT ERRORS OF OUR MODIFIED RANSAC COMPARED TO STANDARD RANSAC, COMPUTED USING 100 RUNS OF THE SCENE REGISTRATION PROBLEM IN FIG. 6.

	Run time [s]	Mean fit [m]
RANSAC	13.68	$6.40 \cdot 10^{-3}$
Modified RANSAC	0.88	$6.31 \cdot 10^{-3}$

1) Model representation: The object models given are taken from the KIT database of full textured CAD models generated using a triangulation scanner and a high resolution camera [34]. We have acquired physical copies of a subset of the objects, which we use for testing. Since ECV processing requires images for capturing the appearance, we render the objects from four different views. The extracted ECV features from a view are backprojected to the 3D model shape, after which the context descriptors are built for that view. During pose estimation, we use the view with the best match in the scene, if such a match exists.

2) Color calibration: An important practical consideration for this experiment is the fact that there may be discrepancies between the color representations in the provided object models and the scenes captured in our setup. This is due to the fact that the object/scene models are captured under different lighting conditions. Since we are not interested in attempting to copy the illumination conditions used when capturing the objects, we devised a simple solution for aligning the color spaces.

We assume that an RGB color triplet $\mathbf{c}_{KIT}$ from a given KIT object model has undergone a linear mapping $\mathbf{A}$ up to the point where it is observed as $\mathbf{c}_{SDU}$ by our sensor:

$$\mathbf{A}\mathbf{c}_{KIT} = \mathbf{c}_{SDU} \tag{5}$$

By estimating $\mathbf{A}$, which is a 3-by-3 matrix, we can transform the color values of the stored object models such that they align with our color conditions. We estimate $\mathbf{A}$ once, simply by labeling a set of corresponding pixel RGB values between an object model and a scene from our laboratory containing that object. $\mathbf{A}$ has nine free parameters, and we have used ten color pixels for estimating it. The resulting color calibration matrix gives the required color space alignment for obtaining valid correspondences using ECV context descriptors.

In Fig. 7, we show pose estimation results for two objects. For completeness, we show both the original and the color calibrated version of an object, which is used during estimation. Note how a fairly large change in color representation is produced by the calibration routine. The estimated object pose is accurate enough for robotic manipulation via grasping.

V. CONCLUSIONS AND FUTURE WORK

We have presented a system for accurately estimating the alignment pose between two 3D models. Our model description is based on both appearance and shape data. Using an Early Cognitive Vision system, we split the input datum into separate representations in the edge and in the

Fig. 7. Pose estimation results from our setup. Top: input Kinect scene. Middle: input textured KIT objects (leftmost instances) and their color calibrated versions (rightmost instances). Bottom: estimation results for both objects, aligned objects overlaid in red.

texture domains. By defining context descriptors on top of these visual modalities, we achieve a highly discriminative interpretation of a scene.

For utilizing our representation in an efficient manner, we have presented a RANSAC-based estimation procedure, which is based on fast rejection of outlier correspondences using low-level geometric constraints. Our experiments indicate that this makes the estimation process almost 15 times faster than standard RANSAC, without compromising estimation accuracy.

We have shown by quantitative evaluations that our descriptors perform almost twice as good as state of the art descriptors in the image domain and in the shape domain, thus making our context-based descriptors highly discriminative. For ten very different object classes, we achieve an average correspondence score of 27 %.

Finally, we have shown the practical usability of our system by accurate alignment results in two real setups. The system is currently being used in robotic grasping applications at our institute, and in the future we plan on using our system for accurate calibration between multiple cameras and robots.

In future works, a thorough investigation should be done in order to optimize the formulation of the context descriptor for ECV features. In this work, we have chosen a straightfor-

ward histogramming approach, incorporating simple angular and RGB relations. Although this has proven efficient, the context descriptor can undoubtedly be improved by alternative geometry- and appearance-based differential metrics, possibly using the local magnitude, orientation and phase which are right now used only for local image structure classification.

REFERENCES

[1] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics (TOG)*, vol. 21, pp. 807–832, Oct. 2002.

[2] D. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision (ICCV)*, vol. 2, pp. 1150 –1157, 1999.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, pp. 509–522, Apr. 2002.

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

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005.

[6] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *IEEE European Conference on Computer Vision (ECCV)*, May 2006.

[7] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, pp. 433–449, May 1999.

[8] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3D object recognition from range images using local feature histograms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. II–394–II–399, 2001.

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

[10] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3212 –3217, May 2009.

[11] P. Bariya and K. Nishino, "Scale-hierarchical 3D object recognition in cluttered scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1657 –1664, June 2010.

[12] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 998 –1005, June 2010.

[13] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Proceedings of the third Eurographics symposium on Geometry processing*, 2005.

[14] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3384–3391, 2008.

[15] 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.

[16] F. Pilz, N. Pugeault, and N. Krüger, "Comparison of point and line features and their combination for rigid body motion estimation," in *Statistical and Geometrical Approaches to Visual Motion Analysis*, vol. 5604, pp. 280–304, 2009.

[17] R. Detry, N. Pugeault, and J. Piater, "A probabilistic framework for 3D visual object representation," *IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)*, vol. 31, no. 10, pp. 1790–1803, 2009.

[18] G. Kootstra, M. Popović, J. A. Jørgensen, K. Kuklinski, K. Miatliuk, D. Kragic, and N. Krüger, "Enabling grasping of unknown objects through a synergistic use of edge and surface information," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 10, pp. 1190–1213, 2012.

[19] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[20] M. Felsberg and G. Sommer, "The monogenic signal," *IEEE Transactions on Signal Processing*, vol. 49, pp. 3136–3144, December 2001.

[21] M. Felsberg, S. Kalkan, and N. Krüger, "Continuous dimensionality characterization of image structures," *Image and Vision Computing*, vol. 27, pp. 628–636, 2009.

[22] N. Pugeault, F. Wörgötter, and N. Krüger, "Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics," *International Journal of Humanoid Robotics (Special Issue on Cognitive Humanoid Vision)*, vol. 7, no. 3, pp. 379–405, 2010.

[23] D. Nistér, "Preemptive ransac for live structure and motion estimation," *Machine Vision and Applications*, vol. 16, no. 5, pp. 321–329, 2005.

[24] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *DAGM Symposium*, pp. 236–243, 2003.

[25] O. Chum and J. Matas, "Matching with PROSAC — progressive sample consensus," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 220–226, 2005.

[26] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, pp. 331–340, 2009.

[27] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision (IJCV)*, vol. 2, no. 60, pp. 91–110, 2004.

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

[29] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions Pattern Analysis and Machine Intelligence (PAMI)*, vol. 14, pp. 239–256, Feb. 1992.

[30] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision (IJCV)*, vol. 13, pp. 119–152, Oct. 1994.

[31] 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)*, pp. 1817–1824, May 2011.

[32] H. Shum and R. Szeliski, "Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment," *International Journal of Computer Vision (IJCV)*, vol. 36, no. 2, pp. 101–130, 2000.

[33] M. Brown and D. Lowe, "Recognising panoramas," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1218–1225, 2003.

[34] 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.