Efficient Multi-View Object Recognition and Full Pose Estimation

Alvaro Collet Siddhartha S. Srinivasa

***Abstract*— We present an approach for efficiently recognizing all objects in a scene and estimating their full pose from multiple views. Our approach builds upon a state of the art single-view algorithm which recognizes and registers learned metric 3D models using local descriptors. We extend to multiple views using a novel multi-step optimization that processes each view individually and feeds consistent hypotheses back to the algorithm for global refinement. We demonstrate that our method produces results comparable to the theoretical optimum, a full multi-view generalized camera approach, while avoiding its combinatorial time complexity. We provide experimental results demonstrating pose accuracy, speed, and robustness to model error using a three-camera rig, as well as a physical implementation of the pose output being used by an autonomous robot executing grasps in highly cluttered scenes.**

I. Introduction

There has been recent renewed interest ([1–4] to name a few) in enabling mobile manipulators to perform useful tasks in unstructured human environments, like homes and offices. Such environments are particularly challenging due to their dynamic nature and due to high clutter. These characteristics demand speed and accuracy from all components: planning, control, and perception. Motivated by these practical requirements, we demonstrated a single-view object recognition and pose estimation algorithm in [5] that is fast, accurate, and robust to clutter.

With cameras getting better, cheaper, and smaller, multiple views of a scene are often easily available. For example, our robot HERB (Fig. 1) has, at various times, been outfitted with cameras on its shoulder, in the palm, on its ankle-high laser, as well as with a stereo pair. Multiple views of a scene are often desirable, because they provide depth estimation, robustness against line-of-sight occlusions, and an increased effective field of view.

Two standard approaches are popular for converting a single-view vision algorithm to multiple views.

The first, which we term *single-view averaging*, executes the single-view algorithm on each of the images and combines the resulting output, often using machine learning techniques [6] (Fig. 2(a)). This approach scales linearly with the number of images and has the ability to combine many single-view algorithms at the same time. However, it treats the single-view algorithm as a black box: fused information is not fed back to the algorithm for further refinement.

The second, which we term *full multi-view*, combines multiple images by considering a network of cameras as one *generalized camera* [7, 8] which produces a single large aggregate image (Fig. 2(b)). The single-view algorithm is then applied to this generalized image. This approach is optimal: there is no loss of information. However, the large search space created by the generalized image makes iterative algorithms such as RANSAC struggle in the presence of thousands of correspondences, due to the exponential increase in computation time when linearly increasing the number of points to be tested.

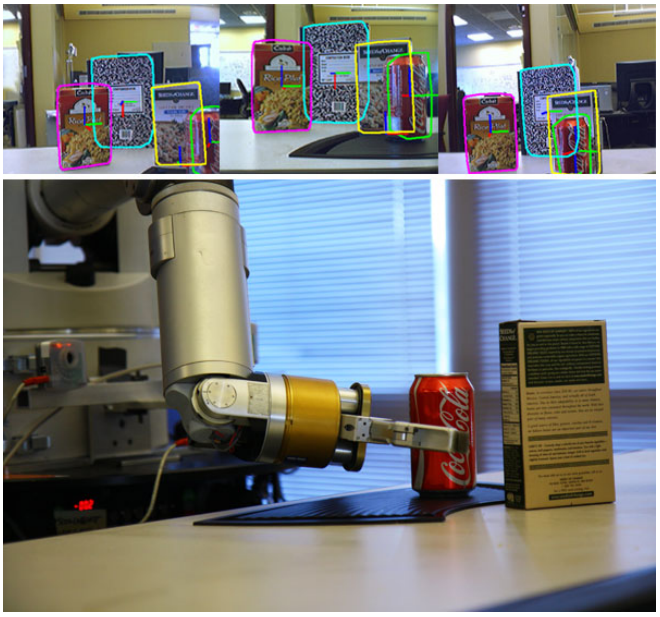

Fig. 1. Object grasping in a cluttered scene using our algorithm. (Top) Scene observed by each camera. Our introspective multi-view algorithm recognizes objects and computes consistent poses across multiple views. Each recognized object is projected back into the images as a coordinate frame plus a convex hull of its projected 3D model. (Bottom) Our robot platform HERB in the process of grasping an object, using only the pose information from this algorithm.

We propose a third approach, which we term *introspective multi-view*, which combines the speed of single-view averaging with the accuracy of the full multi-view approach (Fig. 2(c)). Each individual view is processed first using the single-view algorithm [5], obtaining an initial estimate of objects and their poses. Then, a second stage clusters the output of multiple views, filtering out inconsistent data. Finally, the object pose is re-optimized using a reduced generalized image comprising only of points consistent across all images.

Our multi-step optimization incurs only a slight computational overhead over single-view averaging as a result of its final stage, and only a slight reduction in accuracy over full multi-view as a result of its filtering (Fig. 2(d)). In the general case of complex scenes with clutter, the introspective multi-view algorithm proves to be far superior than single-view and generalized image algorithms, combining the efficiency of the former and the accuracy of the latter.

A. Collet is with The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA - 15213, USA. acollet@cs.cmu.edu

S. Srinivasa is with Intel Labs Pittsburgh, 4720 Forbes Ave., Suite 410, Pittsburgh, PA - 15213, USA siddhartha.srinivasa@intel.com

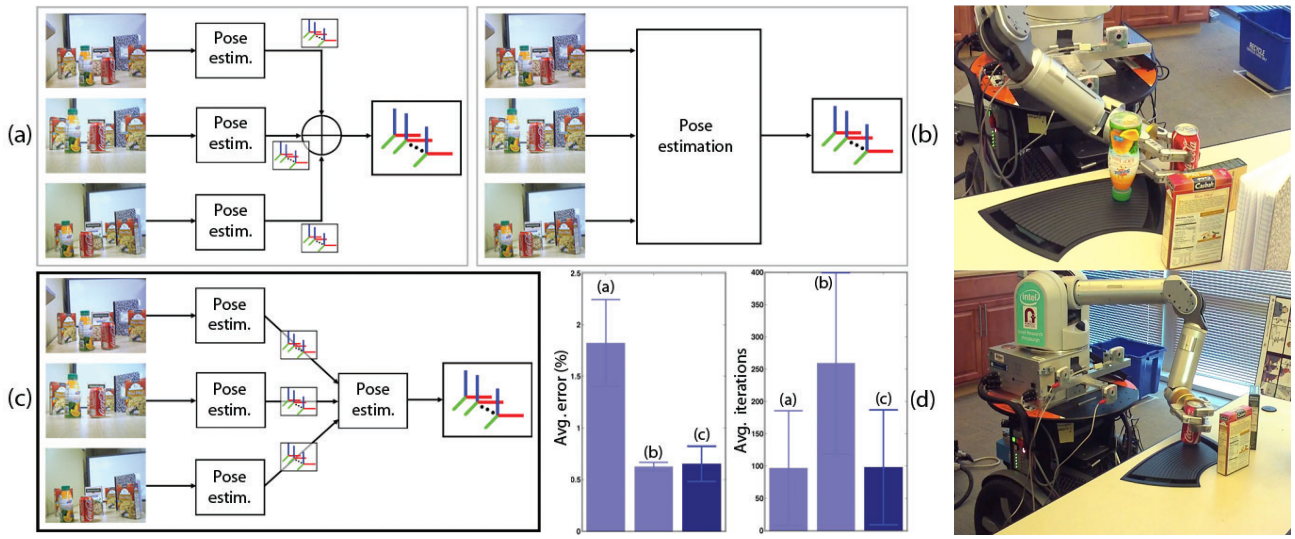

Fig. 2. Multi-view merging alternatives. (a) *Single-view averaging*: Process each view individually using a single-view algorithm and combine their output (e.g. weighted voting, mixture of experts). (b) *Full multi-view*: Embed all views inside a single generalized camera model and use all information to recognize objects and estimate pose. (c) **Our method:** *Introspective multi-view*: Process each view individually, and feed back consistent hypotheses to the pose estimation algorithm for global refinement. (d) Our method has accuracy comparable to (b) and speed comparable to (a). (Right) HERB grasping in clutter using the multi-view algorithm.

II. Single-view recognition and pose estimation

We build upon the single-view algorithm introduced in [5], which this section details.

A. Modeling 3D objects

Objects to be recognized are modeled through an offline learning phase and stored in an object database $\mathcal{O}$. For each object $o \in \mathcal{O}$, a set of images is first taken with the object in various positions and orientations. Reliable local descriptors are extracted from natural features using SIFT features[9]. Using structure from motion [10] on matched SIFT features, information from each training image is merged to produce a sparse 3D model comprising of a set of 3D points $\mathbf{P}_o$ where each point in the set is associated with a local descriptor. Finally, alignment and scale for each model are computed to match the real object dimensions.

B. Recognizing all objects in an image

From a single image, the single-view algorithm finds all known objects in a scene and recovers their accurate 6D pose, even under heavy occlusion and the presence of multiple confusing instances of the same object class.

Our goal is to obtain an *object hypothesis* h comprised of its identity $o \in \mathcal{O}$ and its transformation $T_o \in SE(3)$ with respect to the camera frame, for each object present in the image.

We accomplish this by minimizing the sum of reprojection errors between points in the image and projected points in the model. A novel combination of RANSAC and Mean Shift clustering[11] allows for a real-time solution of the correspondence problem, even with many instances of the same object present.

We repeat the following procedure for each object o in the object database. An example of the robustness to clutter and multiple instances of o is seen in Fig. 3.

1) Extract SIFT features $\mathbf{p}$ from image i and match them against o, obtaining set of correspondences $\mathbf{P}_o^i \leftrightarrow \mathbf{p}_o$.
2) Cluster the 2D image locations of $\mathbf{p}_o$ using Mean Shift.
3) For each cluster c, choose a subset of points and estimate a hypothesis with the best pose according to those points. If the amount of points consistent with the hypothesis is higher than a threshold ϵ, create a new object instance and refine the estimated pose using all consistent points in the optimization. Repeat this procedure until the amount of unallocated points is lower than a threshold, or the maximum number of iterations has been exceeded. This produces a set of hypotheses $\mathbf{h}_c$ for each cluster c.
4) Merge hypotheses from different clusters whose estimates of T_o are similar.
5) Output a reduced set of hypotheses $\mathbf{h}$ for object o.

III. Multi-view recognition and pose estimation

Some multi-view techniques for pose estimation parameterize a network of cameras as a single Generalized Camera[7] and optimize the camera pose over this generalized space by solving the resulting non-perspective PnP (nPnP) problem[12]. While such an approach is perfectly valid, it might not be entirely feasible in real-time if the correspondence problem needs to be taken into account. Another alternative is to combine multiple single-view algorithms via pose verification[13], robust averaging, or weighted voting [14]. These methods avoid the combinatorial explosion that plagues generalized images, but they fail to feedback useful fused information to the underlying algorithm for further refinement.

The single-view object recognition system described in §II analyzes an image and returns several object hypotheses containing information about its identity, its full pose relative to the camera and the set of 3D-2D correspondences consistent with each hypothesis. We now describe our approach to use a single-view object recognition system for a set of images and efficiently fuse the local information to obtain a set of object detections globally consistent with all views.

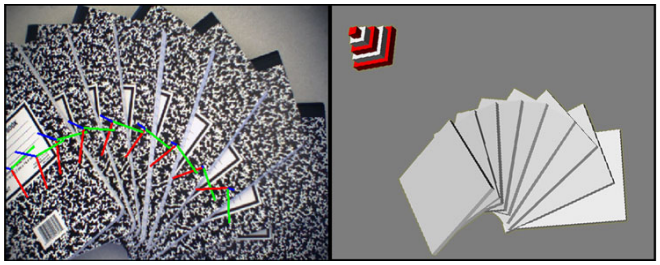

Fig. 3. Recognition of multiple object instances in a (left) Scene observed by the robot's camera, used for object recognition/pose estimation. Coordinate frames show the discovered pose of each object. (right) 3D reconstruction where each object is represented using a simple geometry.

A. Multi-step pose optimization in static camera rigs

We define the multi-view pose optimization algorithm as a multi-step optimization in the sub-image (clusters of points), image and multi-image (clusters of objects) domains.

The algorithm in §II is executed for each individual image i, obtaining an initial hypothesis of all objects in a scene. At the end of this step, each object hypothesis h is linked to its identity and a pose T_o^i relative to camera i. In order to fuse all hypotheses, each object pose is transformed from a camera-centric coordinate system into a common reference frame using the extrinsic parameters $T_i = (R_i, t_i)$ for each camera i producing a global pose $T_o = T_i T_o^i$.

With all objects in the same reference frame, Mean Shift clustering is performed on all poses T_o belonging to the same object class. It is convenient to parameterize rotations in terms of quaternions and project them in the same half of the quaternion hypersphere prior to clustering. As in the single-view case, this produces a set of hypotheses $\mathbf{h}_c$ for each cluster c, with a total number of clusters C. This is often the final output that multi-view integration algorithms offer, a set of reduced set of hypotheses $\mathbf{h}$ obtained by rejecting clusters with less than a certain number of hypotheses to filter out spurious detections or by cluster-merging like in the single-view case. However, it is possible to improve this result further.

The optimal single hypothesis h_c^* for a given cluster c is one that minimize the sum of reprojection errors of correspondences across all images. To accomplish this, we first collect the correspondences for each point $P_j \in \mathbf{P}_o$ in the model across all images, marking 0 if the view i does not contain a corresponding point. This produces a correspondence set that looks like $P_j \leftrightarrow \{p_j^1, p_j^2, 0, 0, \ldots, p_j^M\}$, where M is the total number of views. The optimal single hypothesis is then given by

$$h_c^* = \arg\min_T \sum_{P_j \in \mathbf{P}_o} \sum_{i=1}^{M} \delta_j^i \left[p_j^i - \text{proj}\left(T^i T P_j\right)\right]^2 \quad (1)$$

where $\delta_j^i = 0$ if $p_j^i = 0$ and 1 otherwise.

Alternatively, one can define an analogous optimization in terms of the backprojection error, by tracing the line L_j^i from the camera center to each 2D point p_j^i its distance to the corresponding 3D point P_j. We parametrize a line as $L = (c, v)$, where v is a unit vector indicating the line direction and c is an arbitrary point on that line, *e.g.* the camera center. Using projective geometry, we obtain

$$\bar{v}_j^i = \frac{K_i^{-1} p_j^i}{\| K_i^{-1} p_j^i \|} \quad (2)$$

where K_i is a 3×3 intrinsic camera matrix for view i. Each line L_j^i in a common reference frame is then given by

$$v_j^i = (R_i)^T \bar{v}_j^i \quad c_j^i = -(R_i)^T t_i \quad (3)$$

The distance between a point P and L_j^i is given by

$$d(P_j, L_j^i) = \| \left(\mathcal{I}_{3\times 3} - v_j^i {v_j^i}^T \right) \left(P - c_j^i \right) \| \quad (4)$$

The analogous equation to Eq. 1 that minimizes the sum of backprojection errors is given by

$$h_c^* = \arg\min_T \sum_{P_j \in \mathbf{P}_o} \sum_{i=1}^{M} \delta_j^i \left[d\left(T^i T P_j, L_j^i\right)\right]^2 \quad (5)$$

Additionally, we found it useful to constrain the objects to lie in front of the cameras. Given that v_j^i are vectors from the camera center pointing towards the image plane, ${v_j^i}^T (P - c_j^i) > 0$ for all points P in front of the camera. We incorporate this constraint as a regularizer (with weight $\xi > 0$) in the minimization

$$h_c^* = \arg\min_T \sum_{P_j \in \mathbf{P}_o} \sum_{i=1}^{M} \delta_j^i \left[d\left(T^i T P_j, L_j^i\right) + \xi \left(1 - {v_j^i}^T \frac{(P_j - c_j^i)}{\| P_j - c_j^i \|} \right) \right]^2 \quad (6)$$

Both the reprojection (Eq. 1) and backprojection (Eq. 6) error functions are numerically equivalent when estimating object poses in Euclidean space, so one may choose either one. The reprojection error is usually preferred in the computer vision community because it is invariant to projective transformations, while the backprojection error is meaningless in projective space[15]. In our particular case, working with calibrated cameras in an Euclidean space, we have chosen the backprojection error because it makes our framework more easily extensible to other types of multi-modal data, such as LASER point clouds, which we plan to incorporate in the near future.

If we were considering the general case of pose estimation in multiple views "from scratch", initializing the non-linear minimization from Eq. 6 involves solving the nPnP problem[12] for each iteration of RANSAC, which is computationally very expensive. In our case, we will use the good estimate provided by each cluster centroid as the starting pose.

Two further refinements are necessary to construct a multi-view algorithm from the algorithm described in §II.B. First, the single-view algorithm contains a cluster merging step, designed to fuse information from different clusters within the same image. In the multi-view algorithm, it is also necessary to cluster object poses across different views, and both actions can be integrated in a single clustering step, to merge all clusters within multiple images at the same time. A second issue that requires careful consideration is the clustering radius, i.e. the distance between two object hypotheses to be merged together. Regardless of the choice, it is highly unlikely that a single threshold distance will satisfy

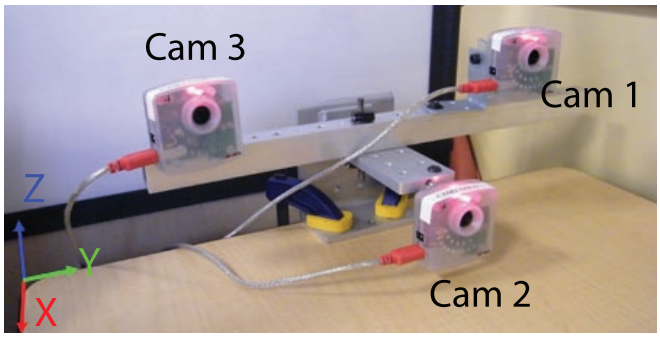

Fig. 4. Three-camera rig used for accuracy tests with coordinate frame indicated on bottom left corner.

all possible cases. If this radius is too small, hypotheses that belong to the same physical object might not be clustered together. If it is too large, a single cluster might envelope several physical objects that are close together (e.g. Fig. 3). The solution is available in the single-view algorithm: RANSAC and a merging step handle well clustering radius issues. The final multi-view algorithm is as follows:

1) Run single-view algorithm on each image without cluster merging.
2) Collect all hypotheses in a common reference frame.
3) For each object class, cluster hypotheses with Mean Shift to obtain sets of hypotheses $\mathbf{h}_c$ for cluster c.
4) For each cluster c, minimize Eq. 6 using all 3D-2D correspondences from all objects that belong to the cluster. Initialize the optimization at the cluster centroid. If number of points consistent with h_c^* is greater than a threshold ϵ_2, consider pose as correct and process next cluster.
5) If ϵ_2 is not met, choose a subset of n points and estimate a hypothesis with the best pose according to those points. If the amount of points consistent with the hypothesis is greater than a threshold $\epsilon_1 < \epsilon_2$, create a new object instance and refine the estimated pose using all consistent points in the optimization. Repeat this procedure until the number of unallocated points is lower than a predetermined amount, or until the maximum number of iterations are exceeded.
6) Merge similar hypotheses using the single-view cluster merging algorithm. Re-optimize using all points.

B. Implementation details

The performance of this multi-view algorithm in real scenes depends largely on the chosen termination conditions for each procedure. Given that each pass refines the solution in search of greater accuracy, it is safe to enforce more stringent constraints as the algorithm proceeds forward. When considering ϵ_2, we are aiming to identify those pose clusters with a strong agreement to avoid further processing. Therefore, choosing a high number of points (e.g. 75% of the total number of points within the cluster) is advisable. ϵ_1 is a fail-safe condition that operates on those clusters with disagreeing poses. We are in an equivalent position to that in the single-view RANSAC step, so a similar value should be chosen. Finally, the last merging step only receives highly refined poses (this algorithm's average translation error is 0.61%), so merging hypotheses should only be performed

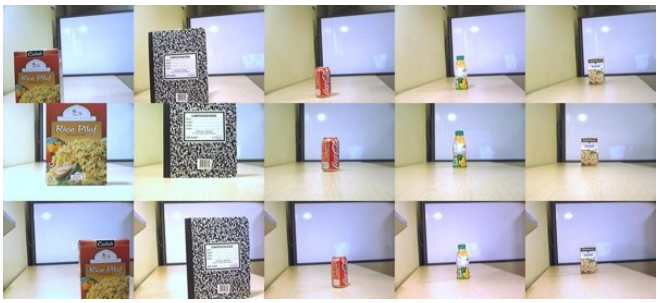

Fig. 5. Examples scenes captured by the rig with Cam 1 (top), Cam 2 (middle), and Cam 3 (bottom). (Col 1) Rice box at 50 cm. (Col 2) Notebook at 60 cm. (Col 3) Coke can at 80 cm. (Col 4) Juice bottle at 1m. (Col 5) Pasta box at 1.2m.

when their positions are closer than $1-2$cm. apart. In addition, multi-view constraints should be enforced to completely eliminate possible false positives in the single-view algorithm, e.g. validating an object hypothesis by requiring that consistent points exist in at least two cameras.

IV. Experiments

Three sets of experiments have been conducted to prove our pose estimation algorithm's suitability for robotic manipulation. The first set evaluates the accuracy of the proposed algorithm in estimating the position and orientation of a given object in a set of images. The second set of experiments evaluates our algorithm's robustness against modeling errors, which greatly influence the accuracy of pose estimation. Finally, the third set uses the pose estimation algorithm alongside a state-of-the-art planning algorithm to grasp objects with a Barrett WAM robotic arm. In all experiments estimate the full 6-DOF pose of objects, and no assumptions are made on their orientation or position. In all cases, the single-view algorithm clusters the scene with a Mean Shift radius of 100 pixels, and chooses subsets of 5 correspondences to compute each RANSAC hypothesis. The maximum number of RANSAC iterations is set to 1000. The multi-view algorithm requires that a pose is seen by at least two views, and that at least 50% of the points from the different hypotheses are consistent with the final pose.

The experimental setup is a static three-camera rig with approximately 10cm baseline between each two cameras (see Fig. 4). Both intrinsic and extrinsic parameters for each camera have been computed, considering camera 1 as the coordinate origin.

A. Pose estimation accuracy

In this set of experiments, we evaluate our system's accuracy over the range most useful in robotic manipulation. The three-camera rig was mounted and calibrated on a flat table(see Fig. 4). Our database is composed of five common household objects of various shapes and appearances. A set of 27 different positions and orientations for each object were gathered, with depths (i.e. distances from the central camera) ranging from 0.4m to 1.2m in 10cm increments, lateral movements of up to 20cm and out-of-plane rotations of up to 45 degrees. 10 images were taken with each camera at each position to account for possible image noise and artifacts, producing 810 images per object and a total of 4050 images.

Fig. 6. Examples of introspective multi-view in complex scenes. (Cols 1-3) depict the recognized poses overlaid on each image. (Col 4) shows a reconstruction of the given scenes in our virtual environment.

TABLE I

AVERAGE ACCURACY TEST. (1) SINGLE-VIEW. (2) ROBUST POSE AVERAGING. (3) INTROSPECTIVE MULTI-VIEW. (4) FULL MULTI-VIEW.

	(1)	(2)	(3)	(4)
TX error (cm)	1.45	1.36	**0.47**	0.46
TX error/dist.	1.80%	1.71%	**0.61%**	0.60%
Rot. error (deg)	6.27	8.11	**5.69**	3.48
Correct det. rate	85.0%	88.3%	**88.3%**	71.9%
False pos. rate	2.78%	0%	**0%**	0%
False neg. rate	13.61%	11.67%	**11.67%**	28.15%
Num iter./view	96.71	96.71	**98.69**	259.05

TABLE II

AVERAGE DISTANCE-NORMALIZED TRANSLATION ERROR WITH VARYING MODEL SCALE.

Model scale	(1)	(2)	(3)	(4)
0.95	4.11%	4.20%	**0.81%**	0.81%
0.97	2.56%	2.65%	**0.68%**	0.62%
0.99	1.86%	1.76%	**0.61%**	0.54%
1.01	2.12%	1.95%	**0.74%**	0.69%
1.03	3.14%	2.90%	**0.98%**	0.94%
1.05	4.72%	4.43%	**1.29%**	1.18%

Table I compares the accuracy of Collet *et al*'s single-view algorithm, robust pose averaging of single views, the proposed multi-view algorithm and a Full Multi-view (Generalized Image) approach. Single-view results show the average performance of [5] over each of the three cameras in our setup. Robust pose averaging computes a weighted mean pose based on the single-view pose hypotheses, using a metric based on the number of consistent points and average reprojection error as a weighting factor. The distance-normalized translation error refers to the absolute translation error divided by the distance with respect to the closest camera. Rotation error is measured as the quaternion angle $\alpha = 2cos^{-1}(q^T q_{gt})$. The correct detection rate counts all pose hypotheses that lie within 5 cm of the true pose. It is important to note that the correct detection, false positive and false negative rates do not necessarily need to add up to 100%, because an algorithm might output a correct and an incorrect pose in the same image.

As we can see in Table I, accuracy is increased threefold using the introspective multi-view scheme with respect to pose averaging, while requiring similar processing time. It is noteworthy that the introspective multi-view and full multi-view, considered a theoretical limit, perform very similarly in terms of accuracy. The low detection rate of the full multi-view algorithm is due to its enormous computational cost, as it often exceeds the maximum number of iterations with no correct detection. The average number of iterations required to detect a single object with a full multi-view approach is three times greater than with any other technique, and its computational complexity grows exponentially with respect to the number of objects in a scene.

B. Robustness against modeling noise

This set of experiments evaluates our proposed algorithm's robustness against modeling inaccuracies. Successful pose estimation in our single-view algorithm is heavily dependent on a good model calibration, specially in terms of scaling, because depth is estimated entirely based on an object's scale. Therefore, extreme care needs to be taken when creating models to set a proper scale, and several tests need to be conducted before a new object model can be incorporated into the robot's knowledge database. For example, a modeling error of 1mm in a coke can (i.e. 1mm larger than its real size), translates into a depth estimation error of up to 3cm at a distance of 1m, large enough to cause problems to the robotic manipulator. On the other hand, having multiple views of the same object enables the use of further constraints in its pose. In particular, if the cameras have been fully calibrated, an "implicit triangulation" takes place during the optimization, with the object drifting to its true position to minimize the global backprojection error, despite the larger error when each view is processed individually.

Table II and Table III showcase the effect of scale errors during the object modeling stage. The proposed multi-view algorithm outperforms every other approach in the presence of modeling noise. It is remarkable that its worst result (with models increasing 5% in scale) outperforms every other approach's best result (with no modeling error, Table I).

TABLE III

AVERAGE CORRECT DETECTION RATE WITH VARYING MODEL SCALE.

Model scale	(1)	(2)	(3)	(4)
0.95	69.7%	71.7%	**80.8%**	59.3%
0.97	82.2%	85.0%	**85.8%**	66.7%
0.99	84.4%	86.7%	**86.7%**	71.1%
1.01	84.2%	88.3%	**88.3%**	70.4%
1.03	74.4%	77.5%	**87.5%**	65.2%
1.05	55.8%	58.3%	**85.0%**	54.1%

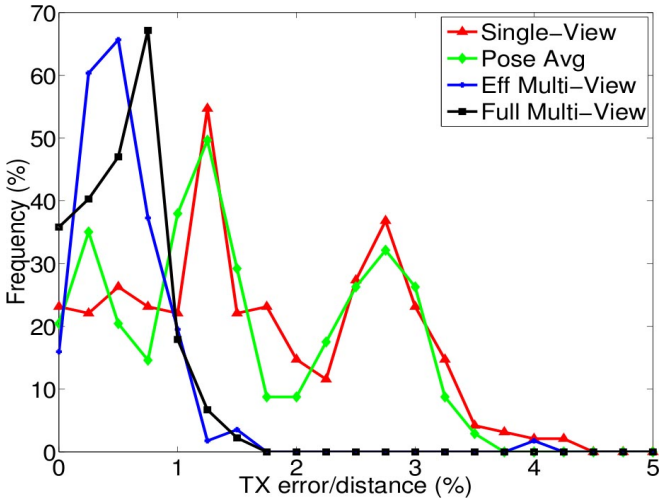

Fig. 7. Distribution of distance-normalized translation errors for different multi-view approaches. Introspective multi-view obtains the most amount of pose hypotheses under 1% error, and global average error of 0.61%

C. Grasping objects

We integrated the multi-view algorithm with a planning algorithm on HERB. The planning algorithm, called the Inverse-Kinematics BiDirectional Rapidly-exploring Random Tree algorithm (IKBiRRT) [16], plans a trajectory for the arm starting from its current configuration and ending at a configuration that places the wrist of the robot at an acceptable location for grasping. Each object that was localized has an associated set of wrist locations that are suitable for grasping. Once the transform of the object is found, the associated wrist locations are input as goal regions for the planner, which then samples from these goal regions as it plans. Note that, for scenes where multiple graspable objects are present, we input all the associated wrist locations for all objects into the planner, which finds a trajectory to reach any one of them. Once the robot completes the trajectory, the fingers are closed and the object is lifted.

In the grasping experiments, one object of each class was placed on a table within the robot's reachable workspace. Before each test, objects were placed in a new arbitrary position and orientation within 10 cm of each other on the table. The robot then planned a trajectory to retrieve each object from the table avoiding the rest and throw it to a trash can. During 20 such scenarios, the robot successfully grasped 98 of the 100 objects (see Table IV), validating the accuracy of the proposed multi-view pose estimation algorithm for robotic manipulation of objects in cluttered scenes.

V. CONCLUSIONS

We have presented and validated an efficient multi-view system for the recognition and registration of common household objects which proves to increase both accuracy and computation time by a factor of three against other multi-view approaches. We have demonstrated that the results are accurate enough for a robot to reach into a cluttered scene and pick up all objects, with a grasping success rate of 98%. We believe that our system provides a crucial capability that will enable mobile manipulators to function and interact in crowded indoor environments with speed and accuracy.

TABLE IV

GRASPING IN CLUTTERED SCENES

	Can	Juice	Rice	Pasta	Notebook	Total
Attempts	20	20	20	20	20	100
Successful grasps	19	19	20	20	20	98

VI. ACKNOWLEDGMENTS

This material is based upon work partially supported by the National Science Foundation under Grant No. EEC-0540865. Alvaro Collet is partially supported by Caja Madrid fellowship. Special thanks to Chris Atkeson and members of the Personal Robotics project at Intel Labs Pittsburgh for insightful comments and discussions.

REFERENCES

[1] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. C. Romea, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and J. M. Vandeweghe, "HERB: a home exploring robotic butler," *Auton. Robots*, vol. 28, no. 1, pp. 5–20, Jan. 2010.

[2] "The PR platform," http://pr.willowgarage.com, 2008.

[3] A. Saxena, J. Driemeyer, and A. Ng, "Robotic Grasping of Novel Objects using Vision," *IJRR*, vol. 27, no. 2, pp. 157–173, 2008.

[4] H. Nguyen, C. Anderson, A. Trevor, A. Jain, Z. Xu, and C. Kemp, "El-e: An Assitive Robot that Fetches Objects from Flat Surfaces," in *HRI, The Robotics Helpers Workshop*, 2008.

[5] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *IEEE ICRA*, Kobe, May 2009, pp. 48–55.

[6] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, 1st ed. Springer, July 2003.

[7] M. D. Grossberg and S. K. Nayar, "A general imaging model and a method for finding its parameters," in *ICCV*, 2001, pp. 108–115.

[8] R. Pless, "Using many cameras as one," *IEEE CVPR*, vol. 2, p. 587, 2003.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, pp. 91–110, 2004.

[10] R. Szeliski and S. B. Kang, "Recovering 3d shape and motion from image streams using non-linear least squares," Robotics Institute, CMU, Pittsburgh, PA, Tech. Rep., March 1993.

[11] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE PAMI*, vol. 17, no. 8, pp. 790–799, 1995.

[12] C.-S. Chen and W.-Y. Chang, "On pose recovery for generalized visual sensors," *IEEE PAMI*, vol. 26, no. 7, pp. 848–861, July 2004.

[13] A. Selinger and R. C. Nelson, "Appearance-based object recognition using multiple views," *CVPR*, vol. 1, p. 905, 2001.

[14] F. Vikstén, R. Söderberg, K. Nordberg, and C. Perwass, "Increasing pose estimation performance using multi-cue integration," in *IEEE ICRA*, Orlando, Florida, USA, May 2006.

[15] R. Hartley and P. Sturm, "Triangulation," 1994.

[16] D. Berenson, S. Srinivasa, D. Ferguson, A. Collet, and J. Kuffner, "Manipulation planning with workspace goal regions," in *IEEE ICRA*, 2009.