Scene Perception and Recognition for Human-Robot Co-operation

Nikhil Somani[1], Emmanuel Dean-León[2], Caixia Cai[1], and Alois Knoll[1,⋆]

[1] Technische Universität München, Fakultät für Informatik,
Boltzmannstrae 3, 85748 Garching bei München, Germany
{somani,caica,knoll}@in.tum.de
[2] Cyber-Physical Systems, Fortiss - An-Institut der Technischen Universität München
Guerickestr. 25 80805 München, Germany
dean@fortiss.org

Abstract. In this paper, an intuitive interface for collaborative tasks involving a human and a standard industrial robot is presented. The target for this interface is a worker who is experienced in manufacturing processes but has no experience in conventional industrial robot programming. Physical Human-Robot Interaction (pHRI) and interactive GUI control using hand gestures offered by this interface allows this novice user to instruct industrial robots with ease. This interface combines state of the art perception capabilities with first order logic reasoning to generate semantic description of the process plan. This semantic representation creates the possibility of including human and robot tasks in the same plan and also reduces the complexity of problem analysis by allowing process planning at semantic level, thereby isolating the problem description and analysis from the execution and scenario-specific parameters.

Keywords: Perception, HRI, Reasoning.

1 Introduction

Industrial robotics, which was hitherto mostly used in structured environments, is currently witnessing a phase where a lot of effort is directed towards applications of standard industrial robots in small and medium sized industries with short production lines, where the scenarios are rather unstructured and rapidly changing. One important challenge for conventional industrial robot systems in these situations is the necessity to re-program the robot whenever the scenario or manufacturing process changes, which requires an expert robot programmer. Standard industrial robot systems also face limitations in their ability to adapt

⋆ The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n 287787 in the project SMErobotics, the European Robotics Initiative for Strengthening the Competitiveness of SMEs in Manufacturing by integrating aspects of cognitive systems.

A. Petrosino, L. Maddalena, P. Pala (Eds.): ICIAP 2013 Workshops, LNCS 8158, pp. 50–59, 2013.

to these environments, and with the complexity of some tasks which seem relatively easier to humans. A partial solution could be to extend the capabilities of the current industrial robots by providing intelligence to these robot systems through perception [14] and reasoning [13] capabilities. This extension of capabilities does not solve the problem completely because these industries typically contain a mixture of tasks, some of which are highly suitable for robots while some others are difficult to model or inefficient for robots and are better suited for humans. This problem stimulates the need for co-operative activities where humans and robots act as co-workers, using the concept of symbiotic Human Robot Interaction (sHRI). This work presents an interface for collaborative human-robot tasks in such industrial environments.

For a robot to be able to work cooperatively with a human, both parties need to be able to comprehend each other's activities and communicate with each other in an intuitive and natural way. In the social robotics and personal robotics communities, meaningful information from human activities is extracted in an abstract or semantic form to achieve this purpose. In an activity containing roles for both human and robot, the level of detail at which the human instructions are specified is important. In several works involving HRI [7,15], human instructions are preferred at an abstract or semantic level. In this case, the scene perception and recognition module is an important component in these intelligent robotic systems. On one hand, the information provided by the perception module is used by reasoning engines to generate an abstraction of the world and learn tasks at this abstract level by human demonstration. On the other hand, the perception module provides scenario specific information which is used by the low-level execution and control modules for plan execution.

The perception problem in this context involves detecting and recognizing various objects and actors in the scene. The objects in the scene consist of workpieces relevant to the task and obstacles, while actors involved are humans, and the robot itself. The most important part of the perception module presented in this work is an object detection, recognition and pose estimation module, which uses 3D point cloud data obtained from low-cost depth sensors and can handle noisy data, partial views and occlusions. The popular approaches for this task can be broadly classified as: local color keypoint [12], local shape keypoint [16], global descriptors [10], geometric [6], primitive shape graph [11]. Global descriptors such as VFH [10] require a tedious training phase where all required object views need to be generated using a pan-tilt unit. Besides, its performance decreases in case of occlusions and partial views. The advantage of these methods, however, lies in their computational speed. Some other methods such as [11], [9] provide robustness to occlusions, partial views and noisy data but are relatively slow and not suitable for real-time applications. In this paper, an extension to the ORR [9] method has been proposed, which enhances its robustness to noisy sensor data and also increases its speed.

To distinguish objects having identical geometry but different color, the Point Cloud is first segmented using color information and then used for object detection. There are several popular approaches for Point Cloud segmentation such as

Conditional Euclidean Clustering [5], Region Growing [4], and graph-cuts based segmentation methods [2]. In this paper, a combination of multi-label graph-cuts based optimization [2] and Conditional Euclidean Clustering [5] is used for color-based segmentation of point clouds.

The major contribution of this article is the integration of the presented perception [14] and reasoning modules [13] in an HRI application. An intuitive interface for instructing industrial robots in unstructured environments typically found in SME's is developed, where scene understanding is a key aspect for HRI and co-operative Human-Robot tasks.

2 Shape Based Object Recognition

The approach presented here is an extension of the ORR method [9], called Primitive Shape Object Recognition Ransac (PSORR) [14]. This approach has two phases : (1) an offline phase where the model point clouds are processed and stored, (2) an online phase where the scene cloud is processed and matched with the models for recognition and pose estimation.

2.1 Primitive Shape Decomposition

This step is very important for the algorithm because the hypothesis generation and pose estimation steps are based on this decomposition. The hypothesis verification step, which is a major bottleneck in most algorithms such as ORR, can also be significantly simplified and sped-up using this decomposition.

Fig. 1. Primitive Shape Decomposition example : (a) original Point Cloud (b) result of Primitive Shape Decomposition

An example of such a decomposition is shown in Fig. 1, where the original scene cloud is shown in Fig. 1 (a) and its decomposition into primitive shapes is shown in Fig. 1 (b).

Hypothesis for primitive shapes are generated by randomly sampling points in the point cloud. Once the hypotheses have been generated, each point in the cloud is checked to determine whether it satisfies the hypotheses.

Each primitive shape has a $fitness\_score$ associated with it, which indicates how well the primitive matches the point clouds, see Eq. 1.

$$fitness\_score = (inliers/total\_points) + K * descriptor\_length \quad (1)$$

where, the first fraction represents the inlier ratio, i.e., the ratio of points which satisfy the primitive shape (*inliers*) to the total number of points in the input cloud (*total_points*), *descriptor_length* represents the complexity of the primitive shape (e.g. the number of values required to represent the shape). The constant K determines the relative weighting of the two factors.

The merging strategy, based on minimum descriptor length (MDL) [8], is a greedy approach where pairs of primitive shapes are selected and merged if the combined primitive shape has a better fitness score than the individual primitive shapes. This continues till there are no more primitive shapes which can be merged.

Planes and cylinders are chosen as primitive shapes for this implementation since they are easy to model and efficient to detect compared to complicated primitives such as ellipsoid or torus. The algorithm, however, is designed to work for any kind of primitive for which a fitness score can be defined according to Eq. 1.

2.2 Hypothesis Generation

An Oriented Point Pair (OPP) (u, v) contains two points along with their normal directions: $u = (p_u, n_u)$ and $v = (p_v, n_v)$. A feature vector $f(u, v)$ is computed from this point pair, see Eq. 2.

$$f(u,v) = \left(\|p_u - p_v\|, \angle\left(n_u, n_v\right), \angle\left(n_u, p_v - p_u\right), \angle\left(n_v, p_u - p_v\right)\right)^T, \quad (2)$$

The central idea in the ORR method is to obtain OPP's from both the scene and model point clouds and match them using their feature vectors. For efficient matching of OPP's, a Hash Table is generated containing the feature vectors from the model point cloud. The keys for this table are the three angles in Eq. 2. Each Hash Cell contains a list of models ($M_i \in M$) and the associated feature vectors. Every feature vector f has a homogeneous transformation matrix F associated with it, see Eq. 3.

$$F_{uv} = \begin{pmatrix} \frac{p_{uv} \times n_{uv}}{\|p_{uv} \times n_{uv}\|} & \frac{p_{uv}}{\|p_{uv}\|} & \frac{p_{uv} \times n_{uv} \times p_{uv}}{\|p_{uv} \times n_{uv} \times p_{uv}\|} & \frac{p_u + p_v}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

where $p_{uv} = p_v - p_u$ and $n_{uv} = n_u + n_v$. Hence, for each match f_{wx} in the hash table corresponding to f_{uv} in the scene, a transformation estimate (T_i) can be obtained, which forms a hypothesis $h_i = \{T_i, M_i\} \in H$ for the model (M_i) in the scene, $T = F_{wx} F_{uv}^{-1}$. The raw point clouds are generally noisy, especially the normal directions. The original ORR method is sensitive to noise in the normal directions and hence, randomly selecting points to generate the feature vectors requires more hypothesis until a good OPP is found. In the PSORR method, every plane in the scene point cloud's primitive shape decomposition is considered as an oriented point (u) with the centroid of the plane as the point (p_u) and the normal direction as the orientation (n_u). The normal directions for these oriented points are very stable because they are computed considering hundreds of points lying on the plane. Therefore, we can use these centroids

instead of the whole cloud to compute and match features, which leads to a significantly less number of hypotheses.

If full views of the objects are available in the scene cloud, the Hash Table for the model cloud can also be computed in a similar fashion considering only the centroids of the primitive shapes. However, in case of partial views or occlusions, the centroid for the scene cloud primitives might not match the model centroids. To handle this, the point pairs for the model cloud are generated by randomly sampling points from every pair of distinct primitive shapes.

2.3 Efficient Hypothesis Verification

Since the model and scene clouds are decomposed into primitive shapes and represented as Primitive Shape Graphs (PSG), hypothesis verification using point cloud matching is equivalent to matching all the primitive shapes in their PSG's. Matching these primitive shapes can be approximated by finding the intersection of their Minimum Volume Bounding Boxes (MVBB's) [1].

The i-th MVBB comprises 8 vertices $v^i_{1,..,8}$, which are connected by 12 edges $l^i_{1,..,12}$ and forms 6 faces $f^i_{1,..,6}$. To find the intersecting volume between MVBB's i and j, the points p^i at which the lines which form the edges of MVBB i intersect the faces of MVBB j are computed. Similarly, p^j are computed. Vertices v^i of the first MVBB which lie inside the MVBB j and vertices v^j of the second which lie inside the MVBB i are also computed. The intersection volume is then the volume of the convex hull formed by the set of points $\left(p^i \cup p^j \cup v^i \cup v^j\right)$.

The fitness score for this match is the ratio of the total intersection volume to the sum volumes of the primitive shapes in the model point cloud. This score is an approximation of the actual match but the speed-ups achieved by this approximation are more significant compared to the error due to approximation.

2.4 Results

Fig. 2 (c) shows an example of the results obtained using the PSORR algorithm, where a partial view of the object is present in the scene cloud, which is much sparser than the model cloud. The algorithm is able to recognize all the object and estimate their poses accurately. The average number of hypotheses required by the PSORR algorithm are nearly 50 times less than the ORR algorithm. Also, the hypothesis verification step is nearly 100 times faster than conventional approaches where point clouds are matched using octrees. Including the additional cost of primitive shape decomposition, the PSORR algorithm is still 5 times faster than the ORR algorithm for the industrial workpieces used in our experiments.

The algorithm has been designed to work with point cloud data and can handle occlusions and partial views. Hence, this data may be from a single frame, combined from several frames over a time interval or fused from several depth sensors in the scene.

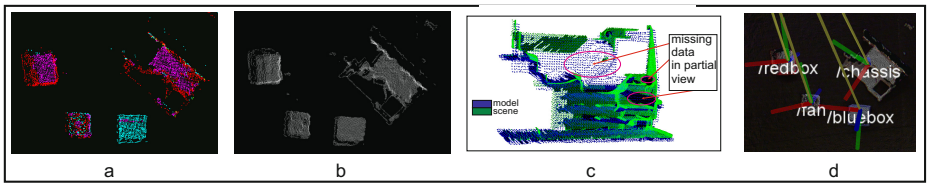

Fig. 2. Example of object recognition using a combination color and shape information: (a) Color Based segmentation (b) Detected Object Clusters (c) PSORR result for partial view of sparse scene cloud (d) Final result of Object Recognition using shape and color information

3 Combining Shape and Color Information

Shape information is often not sufficient for object recognition tasks. For example, some workpieces may have the same shape but different color. A combination of multi-label graph-cuts based optimization [2] and Conditional Euclidean Clustering [5] is used for color-based segmentation of point clouds, followed by cluster recognition and pose estimation using the PSORR method described in Sect. 2.2.

The color based segmentation problem is posed as a multi-label graph-cuts optimization problem. A graph $G = \{V, E\}$ is constructed such that each point in the point cloud is a vertex $v_i \in V$. An edge E_{ij} connects neighboring vertices v_i and v_j. Labels $l_i \in L$ are defined such that each label represents a color. Each l_i is defined by a Gaussian $N(\mu_i, \Sigma_i)$ in the HSV space. Each of these vertices needs to be assigned a label which indicates the color of the object to which the point belongs. The energy term associated with this graph is defined by $D = D_p + D_s + D_l$.

D_p is the data term. It represents the likelihood that the node v_i belongs to the label L_j. D_s is the smoothness term, which represents the energy due to spatially incoherent labels. It can be considered as an interaction term between neighboring nodes, where neighbors prefer to have same labels. D_l is the label swap term. It is an indication of the likelihood of swapping labels for a given vertex. These terms are generally set offline using color models. In this context, the labels which are likely to get mixed up easily (e.g. white and metal) are assigned a higher probability whereas labels which are unlikely to get mixed up (e.g. red and blue) are assigned lower probability.

Fig. 2 shows an example of the results obtained using this approach.

4 Intuitive Interface for Human-Robot Collaboration

The scene perception and recognition algorithm, along with the reasoning module [13] are used to create an interface for human-robot interaction. The perception and reasoning modules help in creating an abstract semantic representation of the world containing objects, actors and tasks. This representation is a key

factor in making the interface intuitive for the user since the user can now communicate with the robot system at an abstract level without the need of numeric parameters.

A mixed reality interface is created using scene perception and reasoning modules, targeted towards human-robot co-operation applications. This interface can be used for teaching process plans at a semantic level (see Fig. 4 (a,b,c)), and execute them in different scenarios without requiring any modifications (see Fig. 4 (d,e,f)). This interface can also be used for executing process plans with both human and robot tasks, see Fig. 4 (g,h,i). Fig. 3 shows an example with the different phases of this interface, where it can be noted that the generated process plan contains semantic names of the objects and not the numeric level data in the form of poses taught to the robot.

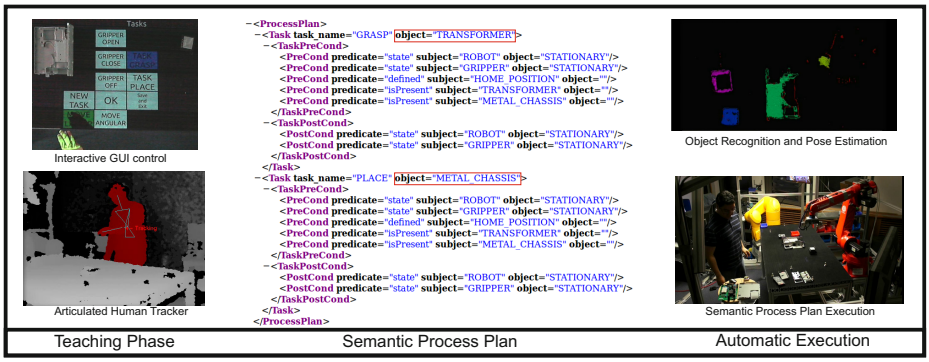

Fig. 3. Overview of Intuitive Interface for Human-Robot Collaboration

4.1 Teaching Process Plans

An articulated Human Tracker is used to recognize the hand positions and use it to control the projected GUI, see Fig. 4 (a). This module enables the user to physically interact with the robot, grab it and move it to the correct position for grasping and placing objects, see Fig. 4 (b-c). The perception module (Sect. 3) detects the objects present in the scene and a reasoning engine associates objects with the taught poses to automatically generate a semantic script of this process plan in STRIPS [3] format, see Fig. 3. The robot system learns process plans and their associated paramters at a semantic level throught this interface. The perception and reasoning module make this learnt process plan independent of the scenario and robot specific details.

4.2 Automatic Plan Execution

The user can place the objects to be assembled anywhere in the working area to begin the plan execution. The system first checks if all pre-conditions for the task

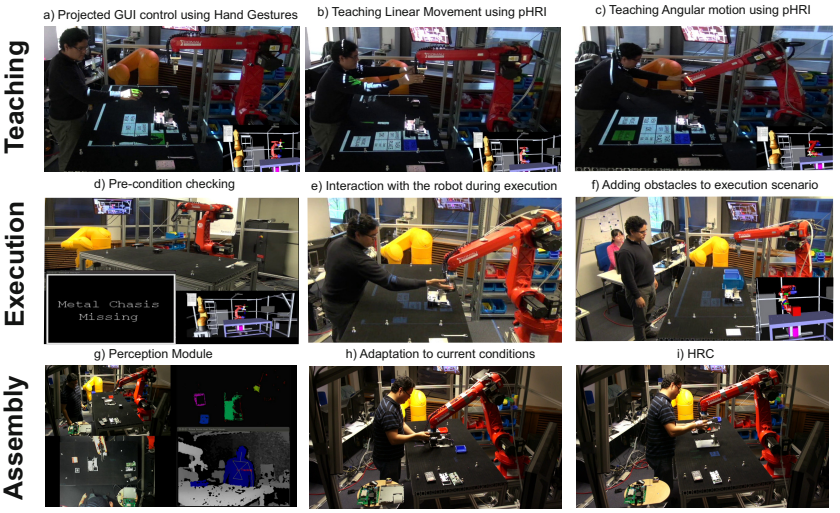

Fig. 4. a,b,c) Teaching Application. d,e,f) Execution and Plan generation of taught Task. g,h,i) HRC in an assembly process.

are satisfied and informs the user in case something is missing, see Fig. 4 (d). The human can physically interact with the robot during the execution and move it by grabbing its end-effector, see Fig. 4 (e). The user can also add obstacles in the path of the robot, which are detected using the perception module and avoided during plan execution, see Fig. 4 (f). All these interactions and changes in the scenario don't require modifications in the process plan script because object positions and obstacles are scenario-specific entities and, like the physical interaction, are handled at the low-level execution. This is the main advantage of decoupling the Problem Space from the Solution Space. The process plan is generated using only information from the Problem Space. The associated execution parameters are loaded on demand. The Perception Module provides the updated information of the current objects in the scene. Therefore, these execution-specific parameters are continuously updated.

4.3 Assembly Task with Human-Robot Co-operation

In this demonstration, we highlight another important advantage achieved using a semantic description of the process plans - possibility of symbiotic human-robot collaboration, which is one of the primary goals of this research. Once the robot is taught the *Pick_And_Place* process plan, it can be instructed to perform this plan on different objects. The application in mind is the assembly of a power converter box. This operation consists of a number of steps, actors and objects which are identified by the perception/reasoning module, Fig. 4 (g), some of which are complex high precision assembly tasks suitable for the human, while some involve lifting heavy objects which are more suitable for the robot. In the situation where

precision assembly is required for a heavy object, a co-operative task is performed where the robot grasps the object and the human guides it by physically grasping the robot end-effector and moving it to the desired place position, Fig. 4 (i). The *Low-Level Execution Engine* switches between motion modalities and control schemes according the current conditions (external perturbations) of the scene, Fig. 4 (h). Thus, in this experiment, we demonstrate the use of this interface for human tasks, robot tasks and co-operative tasks which require both actors. This experiment also highlights that it is relatively easy to understand, edit or even create such a plan from scratch since it is at a semantic level and is abstracted from scenario or execution specific details.

A video illustrating results for the algorithms presented in this paper and its use in the applications mentioned above can be found at:
http://youtu.be/Jgn9NqGKgnI.

References

1. Barequet, G., Har-Peled, S.: Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. J. Algorithms 38, 91–109 (2001)
2. Delong, A., Osokin, A., Isack, H.N., Boykov, Y.: Fast approximate energy minimization with label costs. Int. J. Comput. Vision 96(1), 1–27 (2012), http://dx.doi.org/10.1007/s11263-011-0437-z
3. Fikes, R.E., Nilsson, N.J.: Strips: A new approach to the application of theorem proving to problem solving. Tech. Rep. 43R, AI Center, SRI International (May 1971)
4. Gonzalez, R.C., Woods, R.: Digital Image Processing, 2nd edn. Prentice Hall, New Jersey (2002)
5. Hastie, T., Tibshirani, R., Friedman, J.: 14.3.12 Hierarchical clustering The Elements of Statistical Learning, 2nd edn. Springer, New York (2009) ISBN 0-387-84857-6
6. Hu, G.: 3-D object matching in the hough space. In: Intelligent Systems for the 21st Century Systems, Man and Cybernetics, vol. 3, pp. 2718–2723 (1995)
7. Kirsch, A., Kruse, T., Sisbot, E.A., Alami, R., Lawitzky, M., Brscic, D., Hirche, S., Basili, P., Glasauer, S.: Plan-based control of joint human-robot activities. Künstliche Intelligenz 24, 223–231 (2010)
8. Leonardis, A., Gupta, A., Bajcsy, R.: Segmentation of range images as the search for geometric parametric models. Int. J. Comput. Vision 14(3), 253–277 (1995), http://dx.doi.org/10.1007/BF01679685
9. Papazov, C., Haddadin, S., Parusel, S., Krieger, K., Burschka, D.: Rigid 3D geometry matching for grasping of known objects in cluttered scenes. International Journal of Robotic Research 31, 538–553 (2012)
10. Rusu, R.B., Bradski, G., Thibaux, R., Hsu, J.: Fast 3D recognition and pose using the viewpoint feature histogram. In: 2010 IEEE/RSJ Intelligent Robots and Systems (IROS), pp. 2155–2162 (2010)
11. Schnabel, R., Wessel, R., Wahl, R., Klein, R.: Shape recognition in 3D point-clouds. In: Skala, V. (ed.) The 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2008. UNION Agency-Science Press (February 2008)

12. Sipiran, I., Bustos, B.: Harris 3D: A robust extension of the harris operator for interest point detection on 3D meshes. Vis. Comput. 27(11), 963–976 (2011), http://dx.doi.org/10.1007/s00371-011-0610-y
13. Somani, N., Dean, E., Cai, C., Knoll, A.: Perception and reasoning for scene understanding in human-robot interaction scenarios. In: Proceedings of the 2nd Workshop on Recognition and Action for Scene Understanding at the 15th International Conference on Computer Analysis of Images and Patterns (2013)
14. Somani, N., Dean, E., Cai, C., Knoll, A.: Scene perception and recognition in industrial environments for human-robot interaction. In: Proceedings of the 9th International Symposium on Visual Computing (2013)
15. Zhang, T., Hasanuzzaman, M., Ampornaramveth, V., Kiatisevi, P., Ueno, H.: Human-robot interaction control for industrial robot arm through software platform for agents and knowledge management. In: 2004 IEEE Systems, Man and Cybernetics, vol. 3, pp. 2865–2870 (October 2004)
16. Zhong, Y.: Intrinsic shape signatures: A shape descriptor for 3D object recognition. In: 2009 IEEE Computer Vision Workshops (ICCV Workshops), pp. 689–696 (2009)