

# Human-Robot Collaborative Manipulation Planning Using Early Prediction of Human Motion

Jim Mainprice<sup>1</sup> and Dmitry Berenson<sup>1</sup>

**Abstract**—In this paper we present a framework that allows a human and a robot to perform simultaneous manipulation tasks safely in close proximity. The proposed framework is based on early prediction of the human’s motion. The prediction system, which builds on previous work in the area of gesture recognition, generates a prediction of human workspace occupancy by computing the swept volume of learned human motion trajectories. The motion planner then plans robot trajectories that minimize a penetration cost in the human workspace occupancy while interleaving planning and execution. Multiple plans are computed in parallel, one for each robot task available at the current time, and the trajectory with the least cost is selected for execution. We test our framework in simulation using recorded human motions and a simulated PR2 robot. Our results show that our framework enables the robot to avoid the human while still accomplishing the robot’s task, even in cases where the initial prediction of the human’s motion is incorrect. We also show that taking into account the predicted human workspace occupancy in the robot’s motion planner leads to safer and more efficient interactions between the user and the robot than only considering the human’s current configuration.

## I. INTRODUCTION

Factory automation has revolutionized manufacturing over the last 50 years, but there is still a large set of manufacturing tasks that are tedious or strenuous for humans to perform. Some of these tasks, such as electronics or aircraft assembly, are difficult to automate because they require workers to collaborate in close proximity and adapt to each other’s decisions and motions, which robots cannot currently do. Rather than automating such tasks fully (which may not be possible and/or cost-effective), we believe that human-robot collaboration can enable safe and effective task execution while reducing tedium and strain of the human.

In this paper we address an important step toward human-robot collaboration: allowing a robot and human to safely perform simultaneous manipulation motions in close proximity to one another. Given two sets of tasks  $M$  and  $K$  that the human and the robot could perform, respectively, at a given time, we seek to create a method that selects the robot task and plans the robot motion while 1) minimizing the physical interference between the human and the robot during task execution, and 2) minimizing robot execution time. This involves sensing and predicting the movements of the human, adapting the motion of the robot to avoid the human, and choosing the robot task which interferes the least with the human.

<sup>1</sup> Jim Mainprice and Dmitry Berenson are with the Robotics Engineering Program, Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609, USA. jmainprice@wpi.edu, dberenson@cs.wpi.edu

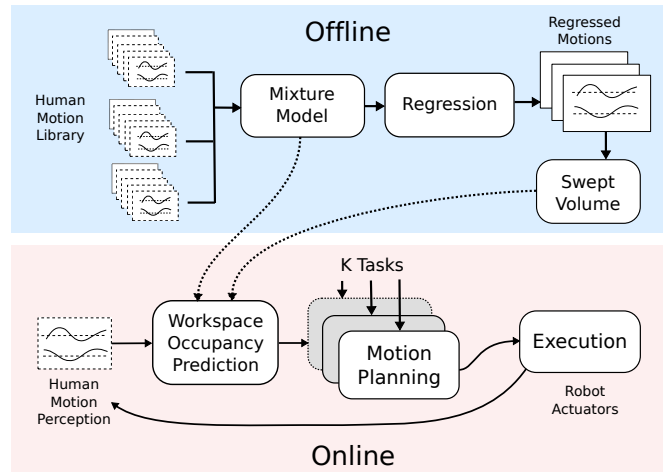


Fig. 1. Data flow of the framework: In the offline stage the human motion library is encoded in a gaussian mixture model and the swept volumes of regressed motions are computed. In the online stage, the workspace occupancy prediction module makes use of the mixture model to produce plans for the robot to minimize interference with the human.

A popular approach to maintaining safety in the presence of moving humans is to interleave planning and execution within a motion planning framework [1], [2], [3]. Humans are treated as dynamic obstacles and the human motion is predicted in the replanning window by using a bounded velocity model. However future motions are unaccounted for, which may lead to inefficiency in the robot behavior. On the other end, research on human-robot collaborative task planning has been focused on taking into account the human explicitly by maintaining a model of affordances [4], [5], and using such analysis at the decision and the motion planning layers. Although these approaches reason on human abilities, they only account for a static model of the human. In this work we propose to account for human motion by learning a task-specific model of human motion and using it to infer the humans future motion and compute safe robot trajectories.

An overview of the framework we have developed to address the above problem is presented in Figure 1. Our method is divided in two phases. In the offline phase, we gather data about how the human moves to perform the set of tasks we are interested in. In the online phase, we use the models learned in the offline phase to predict the human’s motion and simultaneously select a task that interferes the least with the human while producing a robot motion that avoids the human.

An important feature of our framework is early detection

of the human’s intent. This is performed by querying the probabilistic models learned in the offline phase. The result of this query is used to predict the workspace occupancy of the human’s future motions, using the swept volume of the representative motions obtained by regression.

The motion planning component of the framework computes a plan for the  $K$  robot tasks in parallel as the robot moves, and falls back to the best solution at each re-planning step, which gives it an anytime property. Hence as we interleave prediction, planning, and execution, the robot is able to adapt its motion to the human’s intent, minimize interference, and change tasks quickly in order to avoid the human.

This paper focuses on the core algorithms and framework structure that allow safe simultaneous human-robot motion in close proximity. We thus implement, test, and analyze our method in simulation, which is necessary before any human trials are conducted. Given the success of the simulation study presented in this paper, we will pursue physical experiments with human subjects in future work.

The rest of the paper is organized as follows: After discussing related work in section II, we give a detailed overview of the framework in section III. In section IV we present the method for early human motion recognition and prediction of workspace occupancy. Then in section V we present the motion planning component before going over the experimental results in section VI. We end the paper with conclusions and future work.

## II. RELATED WORK

Our work contributes to the field of safe robot manipulation in the presence of humans, it is similar to [6], however their work only considers motions of the human’s hands and does not fall back to the plan of least interference. Older works in the area of safe robot manipulation in proximity to humans, e.g. [7], [8], have led to the characterization of safety criteria. However, though safety is the most important factor of robotic design, it may be desirable to account for other constraints to allow for more comfortable user-interactions and more efficient robot behaviors.

In the closely-related topic of robot navigation in the presence of humans, early works were inspired by the social behavior demonstrated by humans [9]. In those work, the motion planner not only reasons about the human safety, i.e. clearance, but also about how visible the robot is along the robot motion to avoid the effect of surprise [10] leading to more legible navigation plans. Recently this approach was extended to combine planning and more reactive schemes [11], allowing more natural crossings between the robot and the human. Similar to our approach, researchers have explored how to learn models of human motion and use them for online navigation planning [12]. While we believe it is possible to apply our framework to navigation, our approach is intended for manipulation tasks which require planning in a much higher-dimensional space as well as predicting human motion in that space.

Motion planning for robot manipulation in close proximity to humans poses two main challenges: first on the algorithmic techniques to be used and second on the definition of safety and comfort metrics. Recent work has explored computing more human-like motions [13] using a reachability map representation. Other work [5] has focused on considering the human’s safety, visibility, and musculoskeletal comfort directly in the robot configuration space and solving motion planning using sampling-based and stochastic-optimization techniques. The recent work by Dragan and Srinivasa [14] proposes a legibility metric inspired by the psychology of action interpretation applied to robot motion planning using functional gradient optimization. The resulting motion trajectories purposefully deviate from what an observer would expect in order to better convey intent. This work in particular can be seen as the counter part of our approach, where the robot tries to interpret the human’s intent as early as possible.

Simultaneous motion for human-robot collaboration has been studied in terms of hand-over tasks [15], [16], [17]. A motion planning method that allows to share the effort by exploring the combined configuration space of the human and the robot has been proposed in [18]. While hand-over is an important instance of collaboration, the work presented here is meant to apply to a broader class of tasks, although it could be adapted for hand-over tasks as well.

A major part of our framework is early prediction of human motion. Researchers have investigated early recognition of gestures [19] [20], where the goal is to classify a type of gesture being performed by the human. While classifying the type of motion is the first step of our framework, we are also concerned with predicting the human’s future motion and ultimately computing the probability of the human occupying a given voxel in the environment. Our method of learning the Gaussians Mixture Model (GMM) representation of the human’s motion and querying it using Gaussian Mixture Regression (GMR) is similar to [21], although, again, we extend this work to predict the workspace occupancy of the human, as well as integrating the prediction with the robot motion planning.

We rely on recent developments in trajectory optimization for motion planning [1] [22] [23] to compute low-cost robot motion plans quickly. Our framework uses the STOMP algorithm, which has proven effective for the type of manipulator motion planning we consider [23]. Recently, STOMP was adapted to run faster than real-time [3], and we plan to employ this new method in future work.

## III. FRAMEWORK OVERVIEW

Our framework (Figure 1) is composed of two stages: 1) Offline construction of a probabilistic representation of the human workspace occupancy from a motion trajectory library, and 2) online recognition and prediction of human motion and computation of robot motion plans. The online stage also simultaneously computes plans for each possible robot task. The one that minimizes interference with the human is then selected for execution.

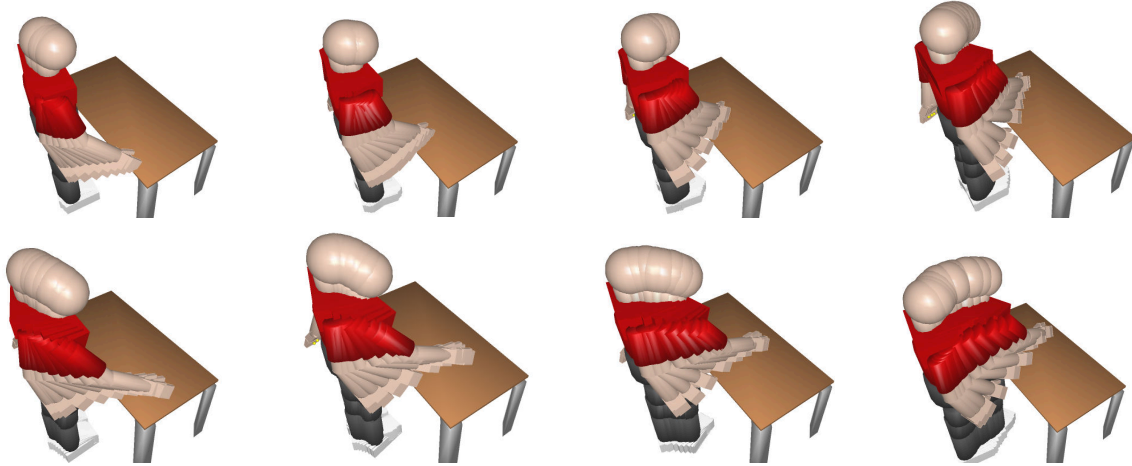


Fig. 2. A motion library gathered using Kinect data is regressed to 8 motion trajectories using GMR. Each motion type corresponds to a different reaching goal position on the table.

### A. Offline phase

A library of motions is first gathered from human demonstration. The type of motion depends on the type of task being executed by the human, here we consider manipulation motion on a table. Each motion in the library is then tagged depending on the intent of the human. For instance, in the manipulation tasks considered in the experiments of section VI, the intent corresponds to the goal position of the hand as the human reaches for an object on the table. In order to perform early motion recognition and predict the subsequent human motion, we fit a GMM to each class of motion. We then use GMR to extract a new motion that best fits the class. Figure 2 shows the regressed motion trajectories used in the experiments. These trajectories are used to compute a swept volume that corresponds to the class which is then stored in a 3D voxel grid.

### B. Online phase

The online phase interleaves prediction, planning and execution. The GMM of each class is first queried with the partial human motion trajectory observed at the current time, yielding likelihood values for each class. The probability of occupancy of a given voxel is then estimated by summing the likelihood of all motion classes whose swept volumes occupy the voxel.

At each replanning step, the cost assessed by the motion planner is updated according to the prediction of human workspace occupancy. We define a penetration cost in the occupancy grid where high cost corresponds to penetration in regions very likely to be occupied by the human's subsequent motion. This cost can be viewed as an interference cost. We compute robot motions that minimize this cost along the robot trajectory using STOMP [23]. We also plan for each task available simultaneously by running STOMP in parallel, one instance per task, and then select the trajectory for the task with the least interference cost at the end of the replanning step.

## IV. EARLY RECOGNITION AND PREDICTION OF HUMAN MOTION AND WORKSPACE OCCUPANCY

Offline, we fit one GMM per class of motion that we are considering in the human motion library. Intuitively, fitting Gaussians to the data set allows us to restrict each class to a small set of parameters, which is straightforward to query.

### A. Gaussian mixture model

Each motion trajectory in our gathered data set is represented by a matrix  $\xi$  composed of  $T$  vectors of dimension  $D$ , with each vector representing a posture.

In gesture recognition, an important problem resides in constructing the feature space, i.e. what values are considered in the posture vector. In [20] the author studies three types of features space, *euclidean* i.e. position of interest points on the kinematic structure of the human, *joint angles*, or *amorphological* i.e. combination of joint angles and relative positions between links, as well as the derivative of the feature vector. We chose joint angles to represent postures, as our aim is to both predict and recognize the motion. Joint angles enable a straightforward reconstruction of the regressed motion with no need for inverse kinematics (which may be difficult due to redundancy). In our experiments 12 DoFs are taken into account: pelvis position and orientation as well as the joint values of the arm and torso.

In the library we define  $M$  classes (which correspond to 8 goal positions in our experiments), the set of motions of class  $m$  is denoted  $C_m$ . Each class consists of  $N$  trajectories. We then fit  $N_g$  gaussians per class; e.g. 20 in our experiments. Thus, the probability density of any posture in a class represented by  $N_g$  gaussians is given by:

$$p(\xi_t) = \sum_{g=1}^{N_g} p(g)p(\xi_t|g),$$

where  $\xi_t$  is the feature vector corresponding to the posture along the motion trajectory  $\xi$  at index  $t$  and  $p(g)$  is the prior

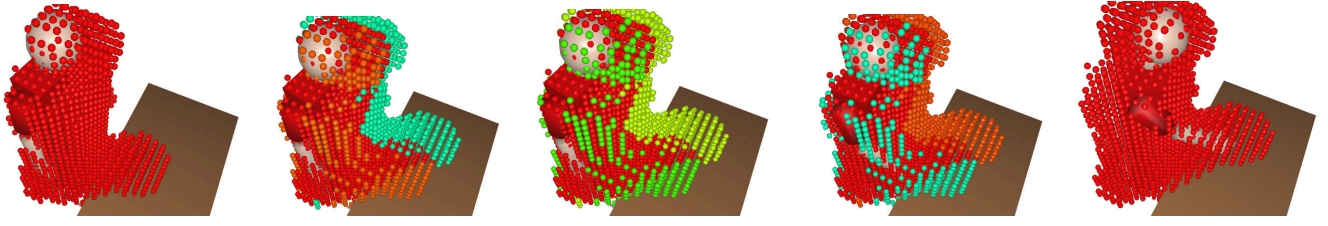


Fig. 3. Evolution of the workspace occupancy prediction stored in a voxel map during a manipulation task. Red spheres correspond to high probability of occupancy and blue to low values. The prediction switches from right to left of the human as he/she progresses towards the goal position.

probability of component  $g$ . The conditional probability for  $g$  is defined as follows:

$$\begin{aligned} p(\xi_t|g) &= \mathcal{N}(\mu_g, \Sigma_g) \\ &= \frac{1}{\sqrt{(2\pi)^D |\Sigma_g|}} e^{-\frac{1}{2}((\xi_t - \mu_g)^T \Sigma_g^{-1} (\xi_t - \mu_g))} \end{aligned}$$

where and  $\{\mu_g, \Sigma_g\}$  are the mean and covariance parameters of the Gaussian component  $g$ . This probabilistic representation of the data set enables two things: first to extract a motion trajectory for each class using GMR, and second, to compute the likelihood that any new motion trajectory belongs to any class.

### B. Training

To fit the GMM to a set of motion trajectories, we have to maximize the likelihood for the mixture parameters. For a given class  $C_m$  comprising of  $N$  motion trajectories, the objective is to maximize the likelihood defined as follows:

$$p(\xi|C_m) = \prod_{n=1}^N \prod_{t=1}^T p(\xi_t^n|C_m)$$

Since the number of considered feature vectors is usually quite high (e.g. 2500 postures per class in our experiments)  $p(\xi|C_m)$  can exceed the machines precision. In order to avoid such situations we used the log-likelihood which is a common procedure for mixture model fitting.

No analytical method exists for maximizing the likelihood. So we use the widely known expectancy-maximization (EM) algorithm which is a simple search technique that monotonically increases the log-likelihood during optimization. The obtained distribution corresponds to a local minimum.

EM consists of two steps. The E step computes the log-likelihood for the parameters  $(p(g), \mu_g, \Sigma_g)$ , and the M steps performs their adjustment through computation of the partial derivatives of the log-likelihood function. We initialize the procedure with an estimate provided by a k-means clustering applied to the data set. The algorithm stops when a negligible improvement of the log-likelihood is attained. The reader may refer to [24] for a more detailed explanation of this procedure.

### C. Query phase

Once the GMMs are fitted for each class, the log-likelihood of all  $C_m$  can be computed as follows:

$$\ln(p(\xi|C_m)) = \sum_{t=1}^T \ln(p(\xi_t|C_m))$$

Note that for the classification problem we would only consider the class with highest log-likelihood. Here we use the likelihood of each class to weigh the voxel occupancy as shown in section IV-E.

### D. Extraction of the swept volume

To obtain a motion reconstructed from the GMMs that represents the predicted trajectory for each class, we use the GMR procedure [25]. More precisely, we apply the technique as established in [21], as it provides a way to reconstruct a general motion for the class.

The swept volume of a given class is computed using the regressed trajectory of the class. To compute the swept volume of the trajectory, we first sample points on the human model 3D surface. We then set the human model to each configuration along the trajectory. The voxel occupancy for one configuration is computed by looking up the voxels that contain the sampled points and marking those as occupied. The sampling has to be dense enough not to miss any voxel. Thus, the swept volume of a given class is computed as the union of the voxel occupancy of discretized configurations along the regressed motion.

### E. Likelihood estimation of workspace occupancy

The motion recognition systems relies on the log-likelihood computed by querying the GMMs which represent the probability of the motions belonging to each class given the partial trajectory observed so far (see Section IV-D). To obtain a prediction  $p(x|\xi)$  for each voxel  $x$  to be occupied by the human motion  $\xi$ , we first retrieve the likelihood  $p(\xi|C_m)$ , the output of the GMM query, of the  $M$  motion classes by exponentiation. We then compute  $p(x|\xi)$  by summing the contribution of the  $M$  motion classes  $C_m$  in the following manner:

$$p(x|\xi) = \sum_{m=1}^M p(x|C_m)p(C_m|\xi)$$

where:

$$p(C_m|\xi) = \frac{p(\xi|C_m)}{\sum_{m=1}^M p(\xi|C_m)}$$

and:



$$p(x|C_m) = \begin{cases} 1 & \text{if } x \text{ is occupied by } C_m \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, this procedure returns a score  $p(x|\xi) \in [0, 1]$  for all voxels that is higher for regions more likely to be occupied by the subsequent human motion, and thus predicts workspace occupancy. This score will be used by the motion planner to avoid colliding with the human and adapt the robot motion to the human's intent. Figure 3 shows the evolution of the estimation of workspace occupancy during the execution of a human motion recorded with the Kinect.

## V. MOTION PLANNING WITH PREDICTED HUMAN WORKSPACE OCCUPANCY

In this section, we describe the motion planning approach based on STOMP. A similar approach has been proposed and studied recently in [3]. Our approach differs from [3] by using workspace occupancy prediction and planning simultaneously for distinct goals.

### A. Planning with predicted human workspace occupancy

STOMP [23] is a trajectory optimizer that iteratively deforms an initial solution by estimating stochastically the gradient in trajectory space. It internally represents the trajectory by an  $m$  by  $n$  matrix, where  $m$  is the number of DoFs and  $n$  the number of waypoints. At each iteration, trajectories are sampled in the neighborhood of the current solution and combined to generate the update. Thus it does not require the analytical gradient of the cost function to be known. The convergence rate to a local minima depends on the standard deviation with which the neighboring trajectories are sampled.

The initial algorithm presented in [23] optimizes a combination of two classical criteria, namely obstacle cost and smoothness cost. The first is estimated by computing a penetration distance in the static obstacles for every waypoint using a signed Euclidean Distance Transformed (EDT). The second is estimated by summing the squared accelerations along the trajectory using finite differencing.

In order to account for the human's intent and minimize interference, we combine a third cost criterion that penalizes configurations penetrating the human predicted workspace occupancy. The penetration cost of a given configuration  $q$  is estimated in the following manner: after sampling points on the robot structure in an initialization phase, we place the robot at  $q$  and evaluate the probability of occupancy of the sampled points. The total cost of  $q$  is then simply the sum of the probabilities of occupancy  $p(x|\xi)$  (see section IV) of the human occupancy-map voxels  $x$ , that contain sampled points.

### B. Planning with multiple goals

Our approach consists of interleaving prediction, planning and execution (see Figure 4). At each replanning step  $K$  STOMPs are run in parallel, one per manipulation task (e.g. pick object1, pick object2, place object1 at p, ...). At replanning step  $n$ , the robot executes the motion planned in

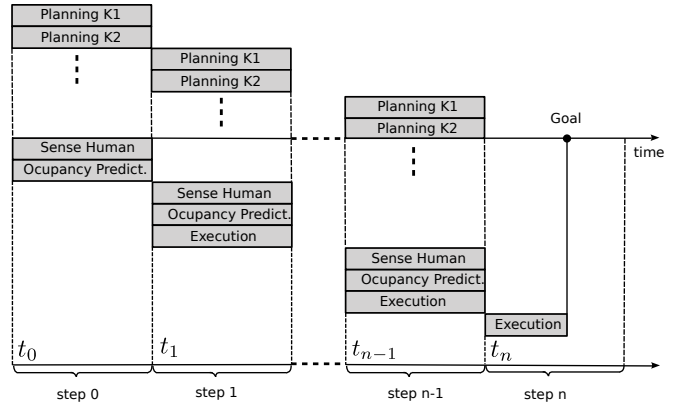


Fig. 4. Interleaving of prediction, planning and execution. The planner uses the prediction that has been established at the previous replanning step.

### Algorithm 1: Multiple goal planning with prediction

```

input   : Workspace  $\mathcal{W}$ 
            $K$  goal configurations  $goals$ 

begin
   $t \leftarrow 0$ 
   $trajs \leftarrow \text{initStraightLines}(goals)$ 
  while taskCompletion() do
     $\xi \leftarrow \text{updateHumanMotion}()$ 
     $X \leftarrow \text{predictVoxelOccupancy}(\xi)$ 
    for  $g \in goals$  do
       $\tau \leftarrow \text{reconnectPrevious}(trajs[g], \tau_{best}, t)$ 
       $trajs[g] \leftarrow \text{STOMP}(\tau, X, \mathcal{W})$ 
     $\tau_{best} \leftarrow \text{getBestTrajectory}(trajs)$ 
     $t \leftarrow \text{execute}(\tau_{best})$ 
end

```

step  $n - 1$ , it also records the current human motion and predicts the human workspace occupancy. This workspace occupancy prediction will be used in replanning step  $n + 1$ .

The complete planner is described in algorithm 1. The planner first starts by initializing straight line trajectories between the current configuration and each goal in `initStraightLines`. It then loops over the three following steps. First, the updated human motion  $\xi$  yields a new voxel occupancy in the `predictVoxelOccupancy` function. This updates the cost function taken into account by STOMP. Then for each goal, the `reconnectPrevious` function adapts the input trajectory in order to cope with the interleaving of the execution step (discussed in section V-C). The planner then launches each STOMP in a different thread. Finally, once each trajectory has been optimized by STOMP, the algorithm executes the trajectory  $\tau_{best}$  that minimizes the overall cost combining human interference, obstacle cost and smoothness.

### C. Interleaving planning and execution when considering multiple goals

When planning with multiple tasks, the execution of the trajectory  $\tau_{best}$  brings the robot away from the trajectories planned for the other tasks. Hence, in order to

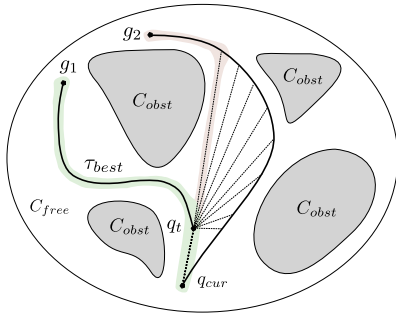


Fig. 5. Depiction of the procedure performed in the `reconnectPrevious` function of algorithm 1 with two goals.

reuse part of the previously optimized trajectories, the `reconnectPrevious` function reconnects the current robot configuration to the previous trajectory by selecting the best feasible solution.

Figure 5 depicts the procedure for two goals  $g_1$  and  $g_2$  where a trajectory has been planned from  $q_{cur}$  to each goal. All straight lines from the future configuration  $q_t$  on  $\tau_{best}$  to the previous trajectory are tested and the trajectory which minimizes cost is kept to provide the next replanning step preformed by STOMP with a good initial input leading to goal  $g_2$ .

## VI. EXPERIMENTS AND RESULTS

We have performed a series of test and experiments on the framework to assess the efficacy of the motion recognition system and the capacity of the framework to minimize interference with the human and switch quickly between robot tasks when necessary. We first evaluate the classification system, then we show the overall efficacy of the framework on two examples. The GMM fitting and GMR algorithms have been implemented in Matlab. The online classifier, STOMP, and a simulated version of the execution framework have been implemented in Move3D [26].

### A. Classification

We first evaluated the performance of the classification component of our framework by gathering a library of  $M = 8$  classes of human motion trajectories each containing  $N = 25$  trajectories using the Kinect sensor. We converted the raw Kinect data into joint-space trajectories using a custom inverse kinematics procedure. The regressed trajectories of each class are presented in Figure 2. In order to verify that this system is adequate for predicting human motions, we evaluated the classification by querying the GMM using leave-one-out testing.

Figure 6 presents the average recognition rate for the 200 motion trajectories as a function of the executed fraction of the motion trajectory. The recognition rate increases as the fraction of the motion trajectory, until reaching a plateau at 92% of the trajectories being correctly classified which occurs at 80% of trajectory execution.

The classifier crosses 50% of good classification with 43% of the motion trajectory and reaches a high percentage of

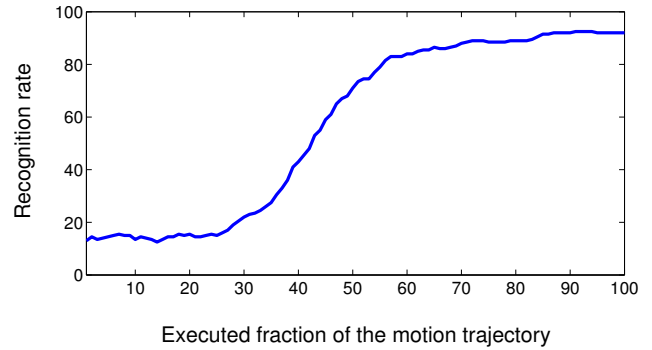


Fig. 6. Leave-one-out testing of the recognition system.

good classification results (approximately 80%) with 60% of the motion trajectory. This shows that the GMM fitting is well suited for early motion recognition and prediction, and can be used for collaborative manipulation. We expect that recognition would decrease as the number of classes taken into account increases. However, this could be compensated with a larger motion trajectory library.

### B. Interference in a manipulation scenario

We study the manipulation task depicted in Figure 7 where the robot and the human reach for two cups set on a table. At each replanning step, we move the human forward along the recorded motion trajectory by a fixed amount, estimate the workspace occupancy and replan the robot motion. We execute 20 replanning steps in total for each run.

A replanning step comprises 50 iterations of STOMP which take approximately 0.5 sec. We plan to make use of the GPU accelerated framework described by [3] to decrease planning time. Given their speed-up description, we can expect to reduce planning time by a factor of 20 to 30.

We compare two versions of the planner. 1) A naive version that computes a path without considering the predicted workspace occupancy. In this case the intermediate path is planned by only considering smoothness and obstacle constraints. 2) A planner that considers the supplementary constraint of minimizing the penetration in the predicted workspace occupancy along the path (our method).

Table I presents the integral cost of the executed robot trajectories considering the final and most accurate prediction of the workspace occupancy. The results are averaged values over 10 runs for 5 human motion trajectories taken from a human motion trajectory library similar to the one described above. The GMMs used in the experiment have been trained leaving those 5 trajectories out.

In Figure 7 where the robot reaches for the the red cup while the human reaches for the blue cup. As one can see taking into account the early prediction of workspace occupancy in the planning phase produces minimal interference with the human and provides good clearance.

This fact is also confirmed by the results presented in table I. Indeed, even though the workspace occupancy predicted with a minimal execution of the human trajectory is inaccur-

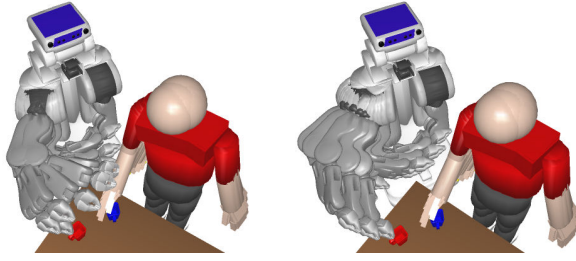


Fig. 7. Left, the motion is planned without considering workspace occupancy, right, considering it as an additional constraint of STOMP.

TABLE I  
AVERAGED COST OF 10 RUNS OF THE MANIPULATION TASK FOR 5  
DISTINCT MOTION TRAJECTORIES OF THE HUMAN

	Cost (basic STOMP)	Cost (least penetration)
Traj. 1	24.41	0.41
Traj. 2	21.47	0.49
Traj. 3	27.52	0.60
Traj. 4	24.44	0.41
Traj. 5	25.13	0.74

rate in the first replanning steps, the robot trajectory produced minimizes the penetration on the final workspace occupancy.

### C. Example with two robot tasks

We now study a slightly different scenario where the human and the robot manipulate objects facing each other across the table (see Figure 8). Two cups are present on the table similarly to the example of Figure 7, and this time the human reaches for the red cup. The robot plans for both tasks (i.e. reaching for both cups), and the goal configurations of the robot are shown on the left side of Figure 8. The trajectories being optimized for each goal configuration are depicted in green and orange. Green for the current  $\tau_{best}$  and orange for the other.

We compare two versions of the algorithm, one that only accounts for the current workspace occupancy of the human (Figure 8.b, 8.c, 8.d) and another that accounts for the predicted workspace occupancy (Figure 8.f, 8.g, 8.h). In the latter case, the early recognition system provides the robot with an initial workspace occupancy prediction that shifts as the human reaches towards the object. This is also depicted in Figure 8 with the same color convention as Figure 3.

Figure 9 presents the evolution of the cost of the two candidate trajectories from the current configuration to the target configuration. This cost is evaluated at the end of each replanning step. The top graph corresponds to using the current observed workspace occupancy version while the bottom graph corresponds to the predicted workspace occupancy version of the algorithm. The integral cost of the trajectory reaching for the blue cup is depicted in blue and in red for the red cup. As shown in Figure 9, the robot switches tasks in step 4 when considering prediction of the workspace occupancy while only switching in step 13 otherwise. Switching task later results in less efficient collaboration as time taken between step 4 and 13 could have been better allocated to perform another task.

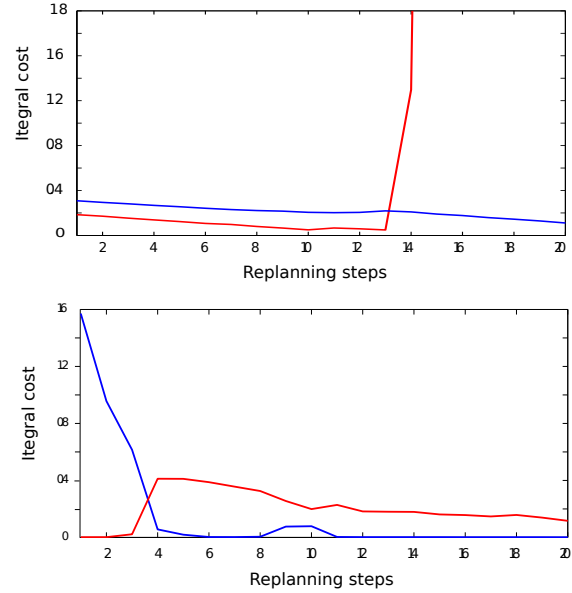


Fig. 9. Integral cost of the two cups example as more replanning steps are executed. Top: only considering current workspace occupancy. Bottom: using workspace occupancy prediction. Red curves correspond to red cups and blue curves to blue cups respectively.

As the ambiguity in the human intent lessens, depicted in Figure 8 by the shift of workspace occupancy prediction, the robot changes the target to the one that minimizes the penetration cost in the predicted workspace occupancy thus leading to more efficient collaborative behavior than produced using only the human's current configuration.

## VII. CONCLUSIONS AND FUTURE WORK

We have presented a framework that allows the human and the robot to perform simultaneous manipulation safely to enable collaborative tasks in close proximity.

This framework is based on early prediction of the human motion. The prediction system, which builds on previous work in the area of gesture recognition, yields a workspace occupancy prediction by computing the swept volume of learned motion trajectories. The motion planner then plans trajectories that minimize a penetration cost in the workspace occupancy and interleaves planning and execution. Plans for multiple tasks can be considered in parallel to select the least-cost task.

Our results show that taking into account workspace occupancy prediction in the motion generation leads to safer, more efficient interactions between the user and the robot than only considering the current configuration. Future work concerns testing this hypothesis on real users with a real PR2. We also intend to investigate a wider range of tasks such as manipulation in shelves or incorporating human navigation.

Finally, our framework considers all available robot actions at the current time to have equal value to the overall process, which may not be the case. To account for varying action costs at the process level, we seek to explore integration with task-level planning for human-robot teams such as [27].

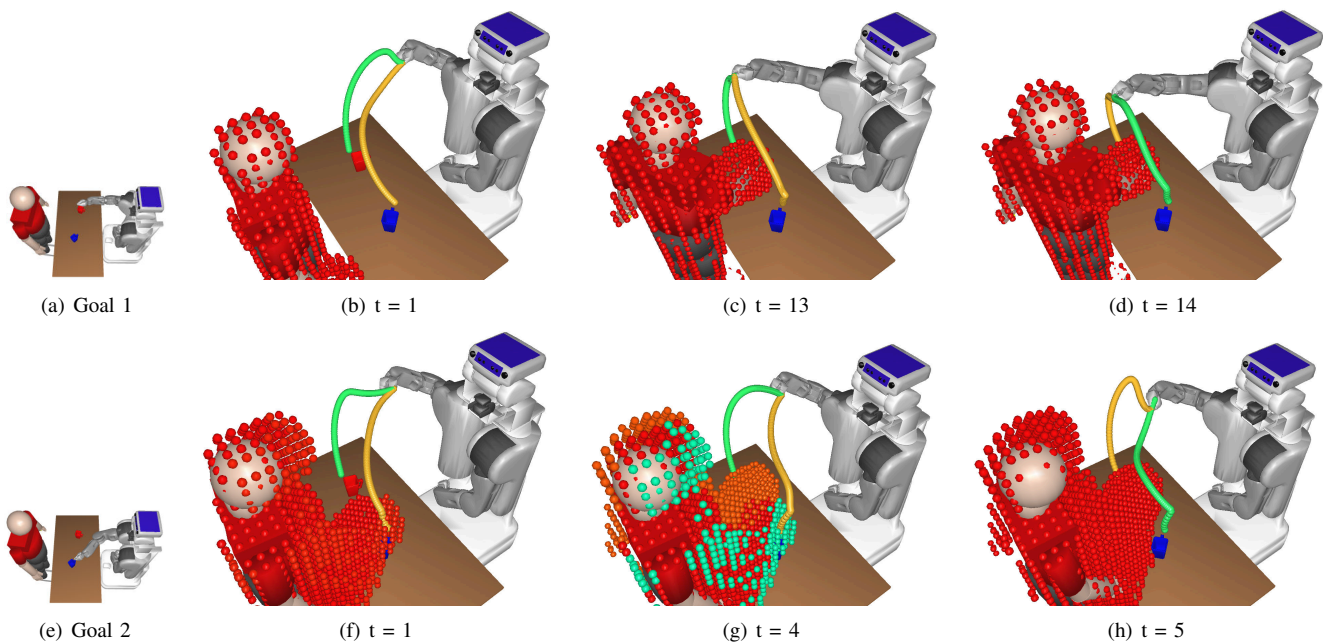


Fig. 8. Optimization of two trajectories and switch between the initial trajectory and the goal. The current trajectory is depicted in green and the other is yellow. The sequence on top (b,c,d) corresponds to planning with the current human workspace occupancy while the bottom sequence (f,g,h) makes use of the predicted human workspace occupancy.

## REFERENCES

- [1] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, 2002.
- [2] T. Fraichard and H. Asama, "Inevitable collision states—a step towards safer robots?," *Advanced Robotics*, 2004.
- [3] C. Park, J. Pan, and D. Manocha, "Real-time optimization-based planning in dynamic environments using gpus," in *ICRA*, 2013.
- [4] A. Pandey and R. Alami, "Mightability maps: A perceptual level decisional framework for co-operative and competitive human-robot interaction," in *IROS, IEEE*, 2010.
- [5] J. Mainprice, E. Akin Sisbot, L. Jaillet, J. Cortés, R. Alami, and T. Siméon, "Planning human-aware motions using a sampling-based costmap planner," in *ICRA*, 2011.
- [6] H. S. Koppula and A. Saxena, "Anticipating human activities using object affordances for reactive robotic response," *RSS*, 2013.
- [7] D. Kulić and E. Croft, "Pre-collision safety strategies for human-robot interaction," *Autonomous Robots*, 2007.
- [8] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for Safe Robots: Measurements, Analysis and New Insights," *International Journal of Robotics Research*, 2009.
- [9] E. T. Hall, "A system for the notation of proxemic behavior," *American anthropologist*, 1963.
- [10] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *Transactions on Robotics*, 2007.
- [11] T. Kruse, P. Basili, S. Glasauer, and A. Kirsch, "Legible robot navigation in the proximity of moving humans," in *Advanced Robotics and its Social Impacts (ARSO), 2012 Workshop on*, 2012.
- [12] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *IROS*, 2009.
- [13] F. Zacharias, C. Schlette, F. Schmidt, C. Borst, J. Rossmann, and G. Hirzinger, "Making planned paths look more humanlike in humanoid robot manipulation planning," in *ICRA*, 2011.
- [14] A. Drangan and S. Srinivasa, "Generating legible motion," *RSS*, 2013.
- [15] M. Huber, M. Rickert, A. Knoll, T. Brandt, and S. Glasauer, "Human-robot interaction in handing-over tasks," in *ROMAN*, 2008.
- [16] A. Edsinger and C. C. Kemp, "Human-robot interaction for cooperative manipulation: Handing objects to one another," in *ROMAN*, 2007.
- [17] M. Cakmak, S. Srinivasa, M. Lee, S. Kiesler, and J. Forlizzi, "Using spatial and temporal contrast for fluent robot-human hand-overs," in *HRI*, 2011.
- [18] J. Mainprice, M. Gharbi, T. Siméon, and R. Alami, "Sharing effort in planning human-robot handover tasks," in *ROMAN*, 2012.
- [19] A. Mori, S. Uchida, R. Kurazume, R.-i. Taniguchi, T. Hasegawa, and H. Sakoe, "Early recognition and prediction of gestures," in *ICPR*, 2006.
- [20] A. Sorel, *Gestion de la variabilité morphologique pour la reconnaissance de gestes naturels à partir de données 3D*. PhD thesis, Université Rennes 2, 2012.
- [21] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Transactions on Systems, Man, and Cybernetics*, 2007.
- [22] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *ICRA*, 2009.
- [23] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *ICRA*, 2011.
- [24] C. M. Bishop, *Pattern recognition and machine learning (information science and statistics)*. Springer, 2006.
- [25] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *arXiv preprint*, 1996.
- [26] T. Siméon, J.-P. Laumond, and F. Lamiroux, "Move3d: A generic platform for path planning," in *Assembly and Task Planning, Proceedings of the International Symposium on*, 2001.
- [27] J. Shah, J. Wiken, B. Williams, and C. Breazeal, "Improved human-robot team performance using chaski, a human-inspired plan execution system," in *HRI*, 2011.