Prediction learning in robotic pushing manipulation

Marek Kopicki, Jeremy Wyatt, Rustam Stolkin
School of Computer Science
University of Birmingham, UK

Abstract— **This paper addresses the problem of learning about the interactions of rigid bodies. A probabilistic framework is presented for predicting the motion of one rigid body following contact with another. We describe an algorithm for learning these predictions from observations, which does not make use of physics and is not restricted to domains with particular physics. We demonstrate the method in a scenario where a robot arm applies pushes to objects. The probabilistic nature of the algorithm enables it to generalize from learned examples, to successfully predict the resulting object motion for previously unseen object poses, push directions and new objects with novel shape. We evaluate the method with empirical experiments in a physics simulator.**

I. Introduction

In early childhood, humans and animals learn models for predicting the interactions with the environment [1]. It seems unlikely that these models comprise an explicit encoding of Newtonian physics, and so must instead rely on a learned relationship between observed actions and their outcomes.

This paper addresses the problem of learning to predict the motion of one body, which results from a forced interaction with another. We have chosen to investigate this problem in the context of robotic "poking" or "pushing" operations, because this includes a large number of unstable manipulations and hence provides interesting situations. However the work is potentially more general.

An algorithm is presented which learns to predict the motions of a rigid object that will result from an applied robotic pushing action. The algorithm does not rely on any understanding or encoding of Newtonian mechanics, but can be trained in simple online experiments in which a robot arm applies random pushes to objects of interest and extracts the resulting motions using a vision system. Properties of objects, and their interactions, are learned as distributions.

Pushing operations are encountered frequently in robotics, but have received relatively little attention in the research community. They are important in that robotic grasping frequently involves a pushing phase, when one finger or jaw of a gripper contacts the workpiece before another. Furthermore, pushing may often be preferable to pick and place type operations if the robot lacks the size or strength necessary to lift an object.

[2] was the first to identify pushing operations as fundamental to manipulation, especially grasping. Mason develops a detailed analysis of the mechanics of pushed, sliding objects and determines conditions required for various 2D motions of a pushed object. [3] attempts to put quantitative bounds on the rate at which these predicted motions occur. [4] developed a method for finding the set of all possible motions of a sliding object, in response to an applied push. More recently, [5] has developed path planning techniques for push manipulation of 2D sliding objects, based on the use of a physics simulator for prediction.

The above work is restricted to planar sliding motions of effectively 2D objects. In contrast, there is comparatively little literature which addresses the far more complex problems of predicting the results of push manipulations on real 3D bodies, which are free to tip or roll. It is possible to use physics simulators to predict the motions of interacting rigid bodies, however this approach is reliant on explicit knowledge of the objects, the environment and key physical parameters. It is therefore not generalizable to new objects or novel situations.

Machine learning approaches have been developed to learn pre-specified binary affordance classes, e.g. rolling versus non-rolling objects [6], or liftable versus non-liftable objects [7]. [8] present experiments where a robot arm coupled to a vision system learns affordances (e.g. rolling or sliding) of various different objects by applying pushes and then observing the resulting motions. This kind of approach is limited, in that affordances learned for a specific object and push action, may not be generalizable to a new object, pose or push direction. Furthermore, although certain primitive classes of motion, e.g. "rolling", may be predicted, such systems cannot predict an explicit 6-DOF rigid body motion for the pushed object.

In contrast, we present a system which can learn to predict the explicit 3D rigid body transformations that will result when an object in an arbitrary orientation is subjected to an arbitrary push. The probabilistic nature of the learning enables generalization to previously unseen push directions and object poses. Furthermore, the system is often able to successfully predict the behaviors of novel objects with previously unencountered shapes.

II. Representing the interaction of rigid bodies

Consider three reference frames A, B and O in a 3-dimensional Cartesian space (see Figure 1). While frame O is fixed, A and B change in time and are observed at discrete time steps $..., t-1, t, t+1, ...$ every non-zero Δt. A frame X

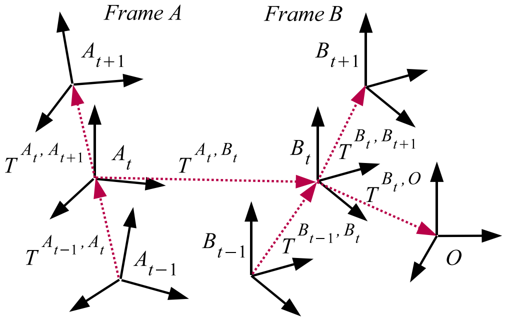

Fig. 1. A system consisting of two interacting bodies with frames A and B in some constant environment with frame O can be described by six rigid body transformations T^{A_t,B_t}, $T^{B_t,O}$, T^{A_{t-1},A_t}, $T^{A_t,A_{t+1}}$, T^{B_{t-1},B_t}, and $T^{B_t,B_{t+1}}$.

at time step t is denoted by X^t, a rigid body transformation between a frame X and a frame Y is denoted by $T^{X,Y}$.

From classical mechanics we know that in order to predict a state of a body, it is sufficient to know its mass, velocity and a net force applied to the body. We do not assume any knowledge of the mass and applied forces, however the transformations of a body, with attached frame B, over two time steps T^{B_{t-1},B_t} and $T^{B_t,B_{t+1}}$ encode its acceleration - the effect of the applied net force. Therefore, if the net force and the body mass are constant, the transformations T^{B_{t-1},B_t} and $T^{B_t,B_{t+1}}$ provide a complete description of the state of a body at time step t in absence of other bodies. A triple of transformations $T^{B_t,O}$, T^{B_{t-1},B_t} and $T^{B_t,B_{t+1}}$ provide a complete description of a state of a body in some fixed frame of reference O which accounts for a constant or stationary environment. Similarly, transformations $T^{A_t,O}$, T^{A_{t-1},A_t} and $T^{A_t,A_{t+1}}$ provide such a description for some other body with frame A.

The state of a system consisting of two bodies with frames A and B in some constant environment with frame O can be described by the six transformations as it is shown in Figure 1, where $T^{A_t,O}$ has been replaced by a relative transformation T^{A_t,B_t}. The transformation $T^{B_t,O}$ can be omitted, if the environment does not affect the motion of the bodies or it is explicitly modeled by one of them.

The prediction problem can now be stated as: given we know or observe the starting states and the motion of the pusher, $T^{A_t,A_{t+1}}$, predict the resulting motion of the object, $T^{B_t,B_{t+1}}$. This is a problem of finding a function:

$$f : T^{A_t,B_t}, T^{B_t,O}, T^{A_{t-1},A_t}, T^{B_{t-1},B_t}, T^{A_t,A_{t+1}} \to T^{B_t,B_{t+1}} \tag{1}$$

Function 1 is capable of encoding all possible effects of interactions between rigid bodies A and B, providing their physical properties and applied net forces are constant in time. Furthermore, it can be learned purely from observations for some fixed time delta Δt. There are two important problems related to relying on such a function:

1) **Limited or no generalization capability.** A function approximating interactions between bodies A and B cannot be used for any other bodies of e.g. different shape or mass. This is because function 1 implicitly encodes information about the surfaces of A and B, which play a critical role in collisions. In this way a slight change of the objects' shape can cause a dramatic deviation of the predicted transformation $T^{B_t,B_{t+1}}$.
2) **Dimensionality problem.** For a rigid body transformation represented as a set of 6 or 7 numbers, the domain of function 1 has 30 or 35 dimensions.

III. Combining Local and Global Information

It is clear that we need to enable generalization of predictions with respect to changes in shape. We also assume quasi-static conditions, i.e. we ignored all frames at time $t-1$. Consider two objects lying on a table top. In Figure 2 there are two situations that are identical except for the shape of the object A, yet it is clear that the same transformation of A's position will lead to quite a different motion for object B. How can we encode the way that the shapes of A and B alter the way they behave? We use a product of several densities to approximate the density over the rigid body transformation given in the function 1.

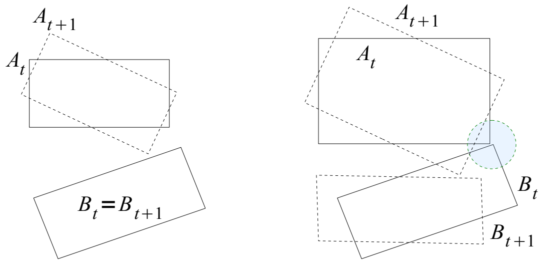

Fig. 2. Two scenes, each with two objects on a table top, viewed from above. Between the two scenes only the shape of A is different. Yet when A moves the resulting transformation $T^{B_t,B_{t+1}}$ will be quite different. This shows that our predictors must take some aspect of the shape of A and B into account.

To do this we approximate two densities, conditioned on local and global information respectively. We define the global information to be the information about the pose, but not the shape, of the whole object. We define the local shape we consider here to be the pose of the surfaces of A and B at the contact point, or the point of closest proximity, between the object and the finger. We model this local shape as a pair of planar surface patches, of limited extent (see Figure 3). Statistically, the greater the starting distance between these local surface patches of A and B, and/or the smaller the magnitude of the transformation $T^{A_t,A_{t+1}}$, the less likely it is that the objects will collide, and hence the less likely it is that the pose of shape B will change between t and $t+1$, or equivalently the more likely that the transformation $T^{B_t,B_{t+1}}$ will be an identity transformation Id. On the other hand, if the local surfaces A and B are close a large portion of possible transformations $T^{A_t,A_{t+1}}$ will cause collisions.

Transformations T^{A_t,B_t}, $T^{A_t,A_{t+1}}$ and $T^{B_t,B_{t+1}}$, observed over many experimental trials for many different

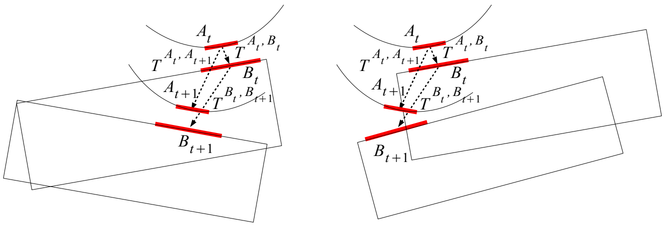

Fig. 3. Two scenes, each with two objects on a table top, viewed from above. Local shapes A and B, transformations $T^{A_t,A_{t+1}}$ and T^{A_t,B_t} are the same in each scene. Still, the transformation $T^{B_t,B_{t+1}}$ is different because local shapes belong to different parts of objects.

objects form a distribution. A particularly useful distribution is a conditional distribution:

$$\{T^{B_t,B_{t+1}}|T^{A_t,A_{t+1}},T^{A_t,B_t}\} \tag{2}$$

While conditional distribution 2 for global frames may become unimodal, for local shapes is highly multi-modal. To see this consider two scenes with two objects, where the initial conditions are identical (Figure 3). Local shapes A and B, transformations $T^{A_t,A_{t+1}}$ and T^{A_t,B_t} are the same in each scene. Still, the transformation $T^{B_t,B_{t+1}}$ is different because local shapes belong to different parts of objects.

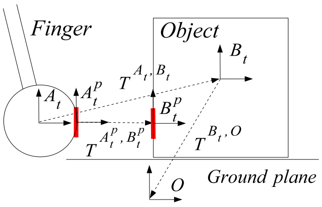

Fig. 4. 2D projection at time t of a robotic finger with global frame A_t, an object with global frame B_t, and a ground plane with constant global frame O. Local frames A_t^p and B_t^p describe the local shape of the finger and an object at their point of closest proximity.

Consider a 2D projection at time t of a robotic finger with global frame A_t, an object with global frame B_t, and a ground plane with constant global frame O (Figure 4). Similarly, local frames A_t^p and B_t^p describe local shapes belonging to a finger and an object. The global conditional density function can be defined as:

$$p(T^{B_t,B_{t+1}}|T^{A_t,A_{t+1}},T^{A_t,B_t},T^{B_t,O}) \tag{3}$$

and similarly a local conditional density function as:

$$p^c(T^{B_t^p,B_{t+1}^p}|T^{A_t^p,A_{t+1}^p},T^{A_t^p,B_t^p}) \tag{4}$$

Because both objects are rigid, $T^{A_t,A_{t+1}} \equiv T^{A_t^p,A_{t+1}^p}$ and $T^{B_t,B_{t+1}} \equiv T^{B_t^p,B_{t+1}^p}$. To predict the rigid body transformation of an object when it is in contact with others we are faced with how to represent the constraints on motion provided by the contacts. We do this using a product of experts. The experts represent by density estimation which rigid body transforms are (in)feasible for each frame of reference. In the product, only transformations which are feasible in both frames will have high probability. For the finger-object scenario a prediction problem can then be defined as finding that $T^{B_t,B_{t+1}}$ which maximizes the product of the two conditional densities (experts) 3 and 4:

$$\max_{T^{B_t,B_{t+1}}} \quad p(T^{B_t,B_{t+1}}|T^{A_t,A_{t+1}},T^{A_t,B_t},T^{B_t,O}) \times \\ p^c(T^{B_t,B_{t+1}}|T^{A_t,A_{t+1}},T^{A_t^p,B_t^p}) \tag{5}$$

The prediction problem cannot be solved using regression approach. Two regression estimates could only be combined linearly since they each make only a single prediction. Without information about the density around each of these predictions there is no ability to find compromise predictions in a principled way. In principle it is possible to fit unimodal densities using regression, but even this approach will lead to failure if the conditional distribution is multi-modal. In this case the conditional distributions are indeed highly multi-modal. Another way of saying this is that since the constraints are clearly highly non-linear the regression approach will fail for even very simple situations.

Starting with some initial state of the finger T^{A_0} and the object T^{B_0}, and knowing a trajectory of the finger $A_1, \ldots A_N$ over T time steps, one can predict a whole trajectory of an object $B_1, \ldots B_N$ by sequentially solving a problem of maximization of the product 5.

There are two major advantages of using such products of densities, e.g. over attempting to directly approximate the function of equation 1:

1) **Efficient movement encoding and learning.** Combining information from both local and global frames, allows objects' properties to be separated into those that are common to many objects and those that are specific to the particular object in question. Common properties (e.g. impenetrability) tend to be encoded in the local surface patches distribution, function 4, whereas the global density function 3 encodes information specific to the workpiece, such as its overall shape. The global density function 3 tends not to require many learning trials to provide accurate predictions, when combined with the local density function 4, which is shared or common to many different objects or situations. Thus this combination provides a movement encoding and learning method which is highly efficient.
2) **Generalization.** Even small differences in a local object surface can cause very different reactions $T^{B_t,B_{t+1}}$ for some given action $T^{A_t,A_{t+1}}$. However, such changes are unlikely to be predicted by a global density function alone. Hence, computing $T^{B_t,B_{t+1}}$ as the maximizer of the product of densities, equation 5, enhances the ability of the system to generalise between different objects and actions, because both local and global densities must simultaneously support the predicted motion hypothesis $T^{B_t,B_{t+1}}$.

IV. Learning as density estimation

We use memory-based learning in which all *learning samples* are stored during learning. The learning samples create a global joint distribution:

$$\{T^{A_t,B_t}, T^{B_t,O}, T^{A_t,A_{t+1}}, T^{B_t,B_{t+1}}\} \tag{6}$$

and local joint distribution:

$$\{T^{A_t^p,B_t^p}, T^{A_t,A_{t+1}}, T^{B_t,B_{t+1}}\} \tag{7}$$

We address $3D$ rigid bodies, subject to 6-DOF transformations, so that distributions 6 and 7 have $4 \times 6 = 24$ and $3 \times 6 = 18$ dimensions respectively. During prediction conditional densities 3 and 4 are created online from learning sample sets (i.e. from distributions 6 and 7).

Consider N D-dimensional sample vectors X_i drawn from some unknown distribution. We would like to find an approximation of this distribution in the form of a density function $p(X)$. Kernel density methods with Gaussian kernels (see e.g. [9]) estimates the density $p(X)$ for any given vector X as a sum of N identical multivariate Gaussian densities centered on each sample vector X_i:

$$p(X) = C_{norm} \sum_{i=1 \ldots N} \exp\left[-\frac{1}{2}(X - X_i)^T \mathbf{C}^{-1}(X - X_i)\right] \tag{8}$$

where a constant $C_{norm} = [N(2\pi)^{D/2}|\mathbf{C}|^{1/2}]^{-1}$ and $\mathbf{C}$ is a $D \times D$ sample covariance matrix. For simplicity, we assume that $\mathbf{C}$ is diagonal. The above equation can be re-written in a new simpler form ([9]):

$$p(X) = \frac{1}{N} \sum_{i=1 \ldots N} \left[\prod_{j=1 \ldots D} K_{h_j}(X^j - X_i^j)\right] \tag{9}$$

where K_{h_j} are 1-dimensional Gaussian kernel functions:

$$K_{h_j}(X^j - X_i^j) = \frac{1}{(2\pi)^{1/2} h_j} \exp\left[\frac{X^j - X_i^j}{h_j}\right] \tag{10}$$

and D parameters h_j are called bandwidth $H \equiv (h_1, \ldots, h_D)$. The bandwidth H is estimated from all distribution learning samples using the "multivariate rule-of-thumb", see [9].

Let us decompose each D-dimensional sample vector X_i into two vectors: K-dimensional Y_i and L-dimensional Z_i so that $X_i \equiv (Y_i, Z_i)^T$ and $D = K + L$. Knowing bandwidth H or equivalently diagonal covariance matrix $\mathbf{C}$ for sample set $\{X_i\} \equiv \{(Y_i, Z_i)^T\}$, we can compute conditional density $p(Z|Y)$ for some given vectors Y and Z using the following two step procedure:

1) Find a set of M weighted samples $\{(Z_i, w_i)\}$ representing a conditional distribution for given vector Y, such that Y_i which corresponds to Z_i lies within some predefined maximum Mahalanobis distance d_{max} to vector Y. Mahalanobis distance d_i between sample vector Y_i and vector Y is defined as:

$$d_i = (Y - Y_i)^T \mathbf{C}_Y^{-1}(Y - Y_i) \tag{11}$$

where diagonal covariance $\mathbf{C}_Y$ is defined as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_Y & 0 \\ 0 & \mathbf{C}_Z \end{bmatrix} \tag{12}$$

Weights w_i are computed from distance d_i as:

$$w_i = \exp[-d_i/2] \tag{13}$$

and normalized for all M weights w_i. Normalized weight w_i can be interpreted as a probability of generating Y_i from a multivariate Gaussian centered at Y with covariance $\mathbf{C}_Y$.

2) Compute conditional probability density $p(Z|Y)$ as:

$$p(Z|Y) = \sum_{i=1 \ldots M} w_i \exp\left[-\frac{1}{2}(Z - Z_i)^T \mathbf{C}_Z^{-1}(Z - Z_i)\right] \tag{14}$$

The density product 5 is maximized using the differential evolution optimization algorithm [10]. This requires the ability to evaluate and sample from each distribution comprising product 5.

All conditional distributions are represented as a weighted set of samples $\{(Z_i, w_i)\}$. Computation of a probability density for some given vector Z is realized as in Equation 14. Sampling consists of a two step procedure:

1) Choose vector Z_i from a set of samples $\{(Z_i, w_i)\}$ using an importance sampling algorithm with importance weights w_i ([11]).
2) Sample from a multivariate Gaussian centered at Z_i with covariance $\mathbf{C}_Z$.

V. Results

We evaluated the prediction algorithm with experiments in a physics simulator. Multiple experimental trials are performed, in which a 5-DOF robotic arm equipped with a finger performs a random movement of length approximately 25 cm towards an object at a random initial pose (Figure 5). In each experiment, learning samples comprising distributions 6 and 7 are stored for a particular object over a series of such random trials. Each experimental trial lasts 10 seconds, while learning samples are stored every 0.1 seconds. Further, new random trials are then generated and the learned distributions are tasked with predicting the resulting motions. Although random trials are independently generated for the learning and prediction phases, the same level of variability in pose and pushing action is used for each phase.

We take the output of a physics simulator to be ground-truth, and compare this with predictions made by our statistical learning method according to an average prediction error E defined as:

$$E = \frac{1}{K} \sum_{k=1 \ldots K} \frac{1}{T} \sum_{t=1 \ldots T} \frac{1}{N} \sum_{n=1 \ldots N} |p_n^1 - p_n^2| \tag{15}$$

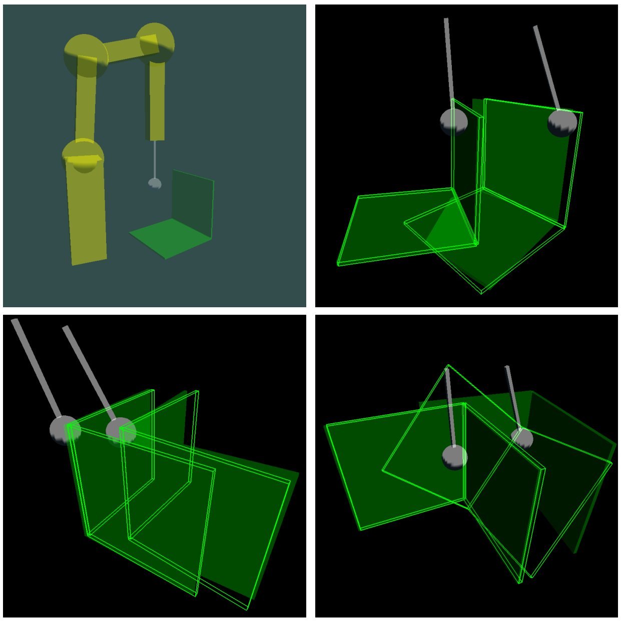

Fig. 5. A 5-DOF robotic arm equipped with a finger performs forward movements towards an object (top left). Object behavior varies depending on the initial object pose and finger trajectory. Physics simulator predictions are rendered as solid, while predictions obtained from our prediction algorithm are rendered as wired. A majority of the algorithm predictions are qualitatively plausible (top right and bottom left). Bottom right panel shows a qualitative error.

where K is a number of experiment trials, T is a number of discrete time steps in each trial (i.e. trial duration), N is a number of pairs of 3D points $\{p_n^1, p_n^2\}$, $|\cdot|$ denotes Euclidean distance between points in a pair. Points p_n^1 are rigidly attached to an object controlled by a physics simulator, while points p_n^2 to an object controlled by the prediction algorithm. All points are randomly generated at the beginning of each trial so that for $t = 1$, $p_n^1 = p_n^2$ for all n.

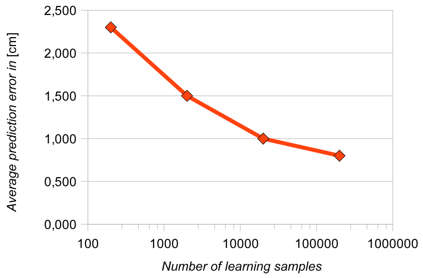

Fig. 6. Average prediction error for a polyflap in a function of a number of learning samples.

In the first experiment a robot pushes a simple symmetric $14cm \times 14cm$ polyflap[1] (Figure 5) placed randomly on a ground plane in arbitrary stable poses.

Figure 6 shows the average prediction error as a function of the number of samples collected during learning (the same for local distribution 7 and global distribution 6). The error decreases as the number of learning samples increases and predictions are reasonably good for just a few thousand learning samples. Even in cases where the prediction errors are large, the majority of predictions are qualitatively plausible, for example, correctly predicting whether a polyflap will slide, tilt or topple (Figure 7).

Polyflap shape modification	Prediction error [cm]
none (learnt shape)	0.76
narrowed by 50%	0.72
widened by 40%	1.3
skewed by 15°	1.16
skewed by 30°	1.26
skewed by 40°	1.35

TABLE I

PREDICTION ERROR IN POLYFLAP EXPERIMENTS.

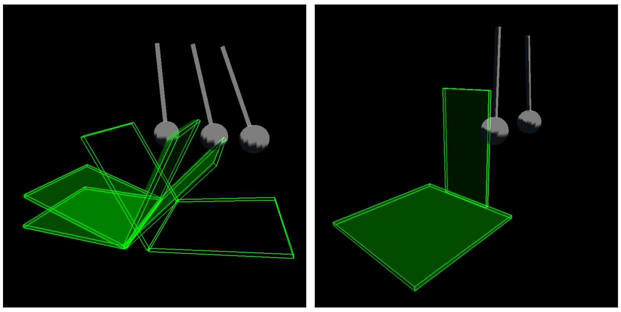

Fig. 7. After the modification of the polyflap shape most of predictions are still qualitatively correct. For example the algorithm predicts that a polyflap tips instead of returning to its initial pose after tilting (left), furthermore it makes no errors if some parts of the surface are removed (right).

In the second experiment we attempted to generalise to novel objects, by using learning samples from the first experiment to predict the trajectory of a new polyflap with a previously unseen shape. We experimented with 5 types of modified polyflap shapes which, together with corresponding prediction error, are collected in Table I. Most predictions are qualitatively correct (Figure 7), however there are more coarse errors compared to the previous experiment.

Box shape modification	Prediction error [cm]
none (learnt shape)	1.68
narrowed by 40%	1.72
widened by 30%	2.15
enlarged by 30%	2.75

TABLE II

PREDICTION ERROR IN BOX EXPERIMENTS.

In a third experiment, we learn on a box ($16cm \times 5cm \times 12cm$ parallelepiped) instead of a polyflap, and try to generalize to predict the motions of distorted boxes that are differently shaped to the one used in learning. We considered 3 types of box modifications which are also compared to the unmodified box shape in Table II. Note that the absolute prediction error is larger than for the polyflap experiment,

[1] Polyflaps are objects consisting of a number of connected flat surfaces. Their behavior can be very complex as compared to e.g. a simple box.

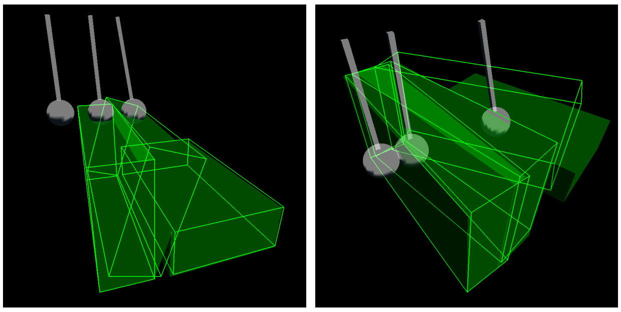

Fig. 8. For a modified box shape version many predictions are correct (left) while if the modified shape extends beyond the learnt one the predictor tends to make more errors (right).

but this is mostly due to the larger box dimensions, and more frequent rotational box movements (see Figure 8).

Prediction	Learning samples	Prediction error [cm]
polyflap	box	2.38
box	polyflap	3.13

TABLE III

PREDICTION ERROR FOR SWAPPED LEARNING SAMPLES.

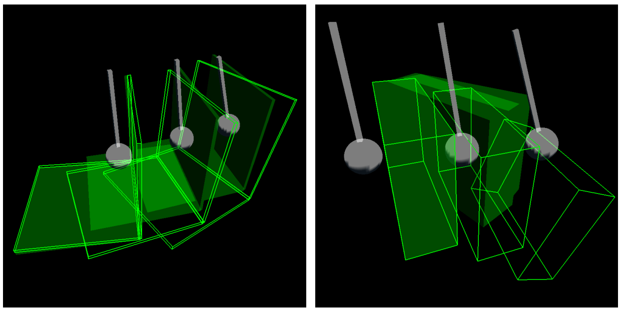

Fig. 9. After swapping learning samples between a box and a polyflap, the majority of predictions are still qualitatively plausible (left). However there is a relatively larger percentage of coarse errors (right).

In the final experiment we attempted to generalise between qualitatively different objects, using box learning samples to predict a polyflap trajectory, and polyflap learning samples to predict a box trajectory (Table III). Again, a majority of predictions are still qualitatively plausible, however there is a relatively larger percentage of coarse errors (Figure 9)

Although the results of these experiments are promising, the algorithm displays errors which are especially visible in the shape generalization experiments and where there are motions involving large amounts of rotation. The major sources of these errors are thought to be:

1) **Density estimation algorithm.** Uniform covariances, used for all kernels, are not completelly adequate for approximating local densities in so many dimensions. Kernel density estimators are unable to handle rank-deficient data [9], whereas such data is clearly present in the introduced distributions.
2) **Correlation problem.** The density product 5 express a correlation (functional relationship) between a finger and an object. While they are correlated if they are in contact, they are no longer strongly correlated when they lose contact following a push. An example is when a polyflap tips over after being pushed by the finger.

VI. Conclusions

We have presented a statistical framework for learning to predict the motions of interacting objects. By decomposing the prediction task into a product of two distributions, each encoding different kinds of information, we have demonstrated a degree of generality in terms of handling variations in shape, poses and actions. We have also shown that it is possible to produce reasonable predictions for the behaviour of a novel shape (e.g. a box), having learned on a quite different one (e.g. a polyflap). This is despite the very small number of densities we use to encode the spatial relationship and shape of the two objects. We are now extending this approach to a product of many densities to give an improved representation of object shape. Future work will also look at using this prediction system for path planning and control during robotic pushing operations.

VII. Acknowledgments

We gratefully acknowledge the support for this work of the EC funded FP7 project CogX, number ICT-215181-CogX.

References

[1] A. Berthoz, *The Brain's Sense of Movement*. Harvard University Press, 1997.

[2] M. Mason, "Mechanics and planning of manipulator pushing operations," *IJRR*, vol. 5, no. 3, pp. 53–71, 1986.

[3] M. Peshkin and A. Sanderson, "The motion of a pushed, sliding workpiece," *IEEE Journal of robotics and automation*, vol. 4, no. 6, 1988.

[4] K. Lynch, "The mechanics of fine manipulation by pushing," in *Proc. IEEE ICRA*, 1992.

[5] D. Cappelleri, J. Fink, B. Mukundakrishnan, V. Kumar, and J. Trinkle, "Designing open-loop plans for planar micro-manipulation," in *Proc. IEEE ICRA*, 2006.

[6] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini, "Learning about objects through action - initial steps towards artificial cognition," in *Proc. IEEE ICRA*, 2003.

[7] L. Paletta, G. Fritz, F. Kintzler, J. Irran, and G. Dorffner, "Learning to perceive affordances in a framework of developmental embodied cognition," in *Proc. IEEE Int. conf. on development and learning*, 2007.

[8] B. Ridge, D. Skocaj, and A. Leonardis, "A system for learning basic object affordances using a self-organizing map," in *Proc. Int. conf. on cognitive systems*, 2008.

[9] D. W. Scott and S. R. Sain, *"Multi-Dimensional Density Estimation"*, pp. 229–263. Elsevier, 2004.

[10] R. Storn and K. Price, "Differential evolution. a simple and efficient heuristic for global optimization over continuous spaces," *J. of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[11] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.