Predicting workpiece motions under pushing manipulations using the principle of minimum energy

Marek Kopicki[1], Rustam Stolkin[1], Sebastian Zurek[1], Thomas Mörwald[2], Jeremy Wyatt[1]
[1]School of Computer Science, University of Birmingham, UK
[2]Automation and Control Institute, Vienna University of Technology, AT

***Abstract*— We are investigating the problem of predicting how objects behave under manipulative actions. In particular, we wish to predict the workpiece motions which will result from simple pushing manipulations by a single robotic fingertip. Such interactions are themselves fundamental components of multi-fingered grasping and other complex interactions. Physics simulators can be used to do this, but they model many kinds of object interactions poorly, being dependent on detailed scene descriptions and parameters, which in practice are often difficult to tune. Additionally, we have previously investigated ways of *learning* to predict, by employing density estimation techniques to learn, from many example pushes, a probabilistic mapping between applied pushing motions and resulting workpiece motions. In contrast, this paper presents an alternative approach to prediction, which does not rely on learning but infers the likelihood of possible workpiece motions by using the simple physics principle of minimum energy. This approach is advantageous in situations where insufficient prior knowledge is available for training our learned predictors. In such situations, possible strategies include either training learned predictors on unrealistic simulation data, or making use of the simple physics approach which requires no training. We show that the second of these strategies performs significantly better, and approaches the performance of learned predictors are trained on observations of real object motions.**

I. Introduction

Pushing operations are encountered frequently in robotics, but have received relatively little attention in the research community. Push manipulations are interesting and challenging in that (especially in 3D problems) they provide a large number of unstable positions. They are also important in that push contacts are fundamental to more complex tasks such as grasping [1]. When a two fingered gripper or a multi-fingered hand approaches a grasp configuration, uncertainty (object geometry and pose subject to sensing accuracy, fingertip pose subject to robot accuracy) means that one finger will typically contact the workpiece before the others, resulting a in a single finger pushing phase before a stable grasp is achieved. Furthermore, any grasp is achieved as a combination of pushing forces from the grasping fingers, and in-hand dexterous manipulation motions are essentially the (non-linear) superposition of the effects on a workpiece of pushing motions due to each of the contacting fingers.

Our previous work [2][3] has presented and compared several algorithms which can learn to predict the motions of a rigid object that result from an applied robotic pushing action. These algorithms do 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. Distributions are important, firstly because they cope with uncertainty of many kinds, and secondly because they enable the opinions of multiple "expert" predictors to be meaningfully combined by a simple product of densities.

This paper presents an alternative approach, in which simple physics principles are used to infer the likelihood of candidate rigid body motions, without the need for learning. This approach is useful, in that it can provide information about the motions of new objects, without having to learn on prior training data for those objects. Furthermore, by expressing the minimum energy principle in terms of a Boltzmann distribution, this simple physics approach can produce, not only a single prediction, but a probability distribution over possible future motions of a workpiece. This means that the opinion of the simple physics predictor can be usefully combined with the opinions of learned density estimators (see above) using the same product of densities scheme. Powerful capabilities for generalization to new objects can now result, by using the simple physics predictor to make overall predictions about gross body motion, while combining with the predictions of learned *local* predictors which have been trained on information about the motions of small parts or surface patches which are common to many objects.

An advantage of the simple physics approach, based on the minimum energy principle, is that it can be applied to previously unencountered objects of arbitrary geometry, and can make relatively robust predictions without exact knowledge of many physical parameters in the scene. In contrast, conventional physics simulation software (e.g. NVIDIA PhysX) might also be applied to these prediction problems, however such techniques are very sensitive to uncertainty in workpiece and scene geometry, and are also dependent on a large number of physical parameters (e.g. frictional constants) which must be very precisely tuned if accurate predictions are to result. In practice, it can be prohibitively difficult or even impossible to tune the parameters of conventional physics simulators such that their predictions match the observed motions of real objects [4]. Furthermore, such simulators make only a single prediction about the future pose of the workpiece, and cannot output a probability distribution over a space of candidate motions. This means that there is no elegant way to combine such physical predictions with our learned predictor techniques, in which we find it useful to combine the opinions

of multiple experts as a product of densities.

The simple physics approach does not generally perform as well as a combination of learned expert predictors which have been trained on *real observations of real objects*. However, the advantage of the simple physics predictor is that it can be used to enhance predictions in situations where insufficient prior knowledge or training data are available for training learned predictors. In such situations our alternative options are: firstly train a combination of learned experts on synthetic training data from simulation environments which do not correspond well to the real world; or secondly, replace the "gross body motion" expert (in the product of experts) with an untrained expert based on simple physics. In this paper we show how the second option, making use of simple physics, significantly outperforms the first option, and can bring the performance of a system (equipped with no prior observations of a new object) closer to the ideal situation, in which a combination of learned experts has been trained on a large body of observations of that object.

The paper proceeds as follows. Section II provides an essential overview of our previous work. We first explain how the motions of the workpiece and pushing fingertip are described by coordinate frames and rigid body transformations between these frames, and show how predictors can be learned from many examples of the rigid body transformations that result from applied pushes. We further explain how objects and their motions can be decomposed, by using several different coordinate frames to encode information about the relative motions of small parts or surface patches of objects. We show how the motions of each of these parts can be learned by multiple *local experts*, and how the opinions of these experts can be meaningfully combined as a product of probability densities.

Section III presents the main focus of this paper, which is an additional or alternative approach to predicting the motions of manipulated objects by making use of basic physics principles. We first describe the principle of minimum energy, and then show how future workpiece poses can be computed as those which minimise the work that was done in reaching them. We further show how a Boltzmann distribution can be used to assign probabilities to a distribution over a space of multiple candidate workpiece motions, and how this description in terms of a distribution enables the opinion of the simple physics predictor to be conveniently combined with the opinions of learned predictors as a product of densities.

Section IV presents experimental test results in which a series of pushes are applied to objects and both learned and physics based predictors are tasked with predicting the resulting motions. Performance is evaluated through a combination of virtual experiments in a physics simulator, and real experiments with a 5-axis arm equipped with a simple, rigid finger, and a vision system which can capture the motions of pushed objects.

For a detailed review of the robotic pushing manipulation literature, and a more detailed exposition of our previous work on learning push predictors via density estimation, see [3].

II. Prediction learning

A. Representations

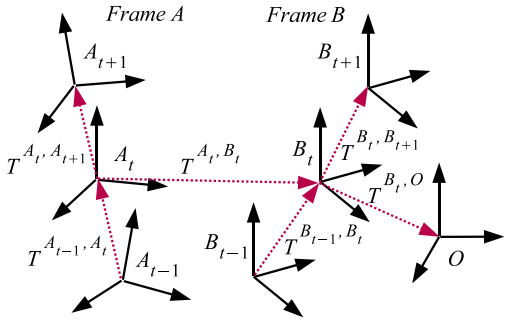

Fig. 1. A system consisting of two interacting bodies with frames A and B in some constant environment with frame O can be described by six rigid body transformations T^{A_t,B_t}, $T^{B_t,O}$, T^{A_{t-1},A_t}, $T^{A_t,A_{t+1}}$, T^{B_{t-1},B_t}, and $T^{B_t,B_{t+1}}$.

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 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 three 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 modelled by one of them.

The prediction problem can 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:

$$f : T^{A_t,B_t}, T^{B_t,O}, T^{A_{t-1},A_t}, T^{B_{t-1},B_t}, T^{A_t,A_{t+1}} \rightarrow T^{B_t,B_{t+1}} \quad (1)$$

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.

In many robotic operations, manipulations are slow, we can assume quasi-static conditions, and it is often possible to ignore all frames at time $t-1$. This conveniently reduces the dimensionality of the problem, giving:

$$f : T^{A_t,B_t}, T^{B_t,O}, T^{A_t,A_{t+1}} \rightarrow T^{B_t,B_{t+1}} \tag{2}$$

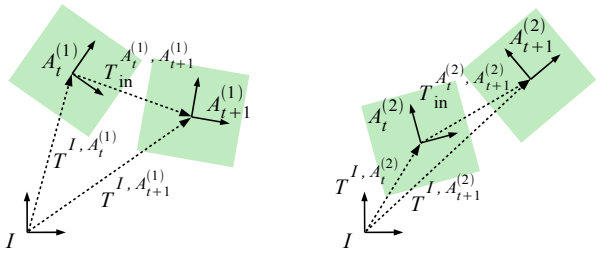

Fig. 2. In the above two scenes a pose change between time step t and $t+1$ as observed in instantaneous object body frame $A^{(1)}$ and the same object in another instantaneous body frame $A^{(2)}$ given inertial frame I are both the same. However because transformations $T^{I,A^{(1)}}$ and $T^{I,A^{(2)}}$ are different, the corresponding transformations in the inertial frame are also different, i.e. $T_{in}^{A_t^{(1)},A_{t+1}^{(1)}} \neq T_{in}^{A_t^{(2)},A_{t+1}^{(2)}}$.

We expect that the behaviour of interacting bodies represented by rigid body transformations as in Figure 1 shares some statistical similarities *independently* on their global poses with respect to some current inertial frame I [3]. Instead of using inertial frame-dependent transformation $T_{in}^{A_t,A_{t+1}}$, one can represent object transformations as observed in the object body frame (see Figure 2). Body frame transformation $T_{body}^{A_t,A_{t+1}}$ is obtained by moving instantaneous frame A, so that at time t it overlaps with inertial frame I. Given some instantaneous object frame A_t at time t, transformation $T_{in}^{A_t,A_{t+1}}$ and because $T^{I,A_{t+1}} = T_{in}^{A_t,A_{t+1}} T^{I,A_t} = T^{I,A_t} T_{body}^{A_t,A_{t+1}}$, one can obtain transformation $T_{body}^{A_t,A_{t+1}}$ in the body frame as follows:

$$T_{body}^{A_t,A_{t+1}} = (T^{I,A_t})^{-1} T_{in}^{A_t,A_{t+1}} T^{I,A_t} \tag{3}$$

Similarly from a given transformation in body frame, instantaneous object frame A_t at t and using Equation 3, one can obtain expression for transformation $T_{in}^{A_t,A_{t+1}}$ in the inertial frame

$$T_{in}^{A_t,A_{t+1}} = T^{I,A_t} T_{body}^{A_t,A_{t+1}} (T^{I,A_t})^{-1} \tag{4}$$

In further discussion we will keep subscripts in while dropping subscripts $body$ assuming that all transformations $T^{X,Y}$ are transformations in the body frame X obtained from $T^{X,Y} \equiv T_{body}^{X,Y} = (T^{I,X})^{-1} T_{in}^{X,Y} T^{I,X}$.

B. Learning global and local experts as density estimation

Prediction learning with using Functions 1 or 2 is limited with respect to changes in shape [3]. Consider two objects lying on a table top. Figure 3 shows two situations that are identical except for the shape of object A. It is clear that the same transformation of A's position will lead to different motions for object B in each case. How can we encode the way in which 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 2.

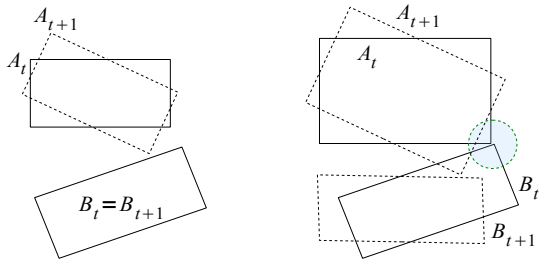

Fig. 3. Two scenes, each with two objects on a table top, viewed from above. Between the two scenes only the shape of A is different. Yet when A moves the resulting transformation $T^{B_t,B_{t+1}}$ will be quite different. This shows that our predictors must take some aspect of the shape of A and B into account.

In the simplest case one can approximate two densities, conditioned on local and global information respectively [3]. We define the global information to be the information about changes of the pose of the whole object. The local information is specified by changes of 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 4).

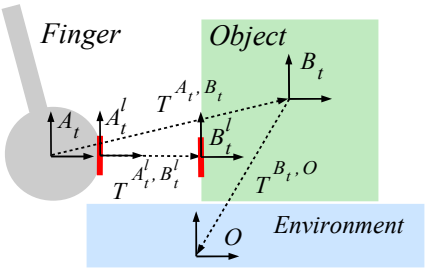

Fig. 4. 2D projection at time t of a robotic finger with global frame A_t, an object with global frame B_t, and a ground plane with constant global frame O. Local frames A_t^l and B_t^l describe the local shape of the finger and an object at their point of closest proximity.

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_l^t and B_t^l describe local shapes belonging to a finger and an object. We define a global conditional density function as [3]:

$$p_{global}(T^{B_t,B_{t+1}} | T^{A_t,A_{t+1}}, T^{A_t,B_t}, T^{B_t,O}) \tag{5}$$

and similarly a local conditional density function as:

$$p_{local}(T^{B_t^l,B_{t+1}^l}|T^{A_t^l,A_{t+1}^l}, T^{A_t^l,B_t^l}) \quad (6$$

To predict the rigid body transformation of an object whe it is in contact with others we are faced with how to represer the constraints on motion provided by the contacts. We do thi using a product of experts [3]. The experts represent by densit estimation which rigid body transforms are (in)feasible fo each frame of reference. In the product, only transformation which are feasible in both frames will have high probability

The only problem is to find relations between transfor mations in the body frame of the local shapes and th corresponding transformations in the inertial frames. For a particular situation shown in Figure 4 from object rigidity and using Equation 3 we have:

$$T^{A_t^l,A_{t+1}^l} = (T^{I,A_t^l})^{-1} T_{in}^{A_t,A_{t+1}} T^{I,A_t^l} \quad (7a)$$

$$T^{B_t^l,B_{t+1}^l} = (T^{I,B_t^l})^{-1} T_{in}^{B_t,B_{t+1}} T^{I,B_t^l} \quad (7b)$$

where I is the inertial frame. $T^{A_t^l,B_t^l}$ can be determined directly from the shape frame:

$$T^{A_t^l,B_t^l} = (T^{I,A_t^l})^{-1} T_{in}^{A_t^l,B_t^l} T^{I,A_t^l} \quad (8)$$

For the finger-object scenario a prediction problem can then be defined as finding that transformation $T_{in}^{B_t,B_{t+1}}$ in the inertial frame which maximizes the product of the two conditional densities (experts) 5 and 6:

$$\max_{T_{in}^{B_t,B_{t+1}}} p_{local}((T^{I,B_t^l})^{-1} T_{in}^{B_t,B_{t+1}} T^{I,B_t^l} | T^{A_t,A_{t+1}}, T^{A_t^l,B_t^l}) \times p_{global}((T^{I,B_t})^{-1} T_{in}^{B_t,B_{t+1}} T^{I,B_t} | T^{A_t,A_{t+1}}, T^{A_t,B_t}, T^{B_t,O}) \quad (9)$$

Starting with some initial state of the finger T^{A_0} and the object T^{B_0}, and knowing a trajectory of the finger $A_1, \dots A_N$ over T time steps, one can now predict a whole trajectory of an object $B_1, \dots B_N$ by sequentially solving a problem of maximization of the product 9.

C. Incorporating information from additional experts

In addition to learning how an object moves in response to a push, it is desirable if we can also incorporate learned information about the inherent tendencies of parts of an object to move in various directions with respect to the environment or any other objects, but regardless of whether it is being pushed or not. This additional information may help when predicting the motions of previously unseen objects, because it provides some prior knowledge about what kinds of motions are possible and which are not.

We can incorporate this additional information by attaching an arbitrary number of additional coordinate frames B^{sn_t} to various parts of the object. We then learn densities for the future motions of each of these frames, conditioned only on their relative pose $T^{E^{S_t^k},B^{S_t^k}}$ with respect to a corresponding

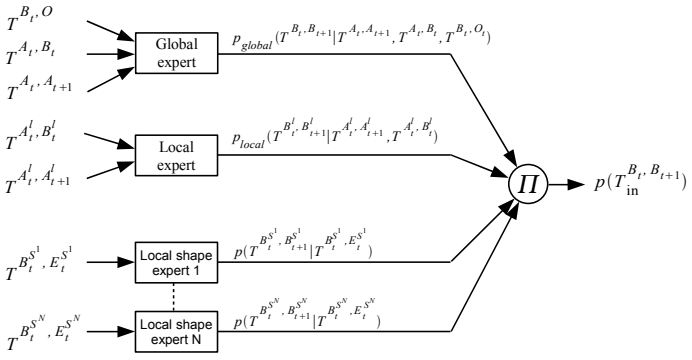

Fig. 5. Inputs and outputs of learned prediction system. The 2-expert approach can be extended to include opinions from multiple local shape experts represented by coordinate frames S^N.

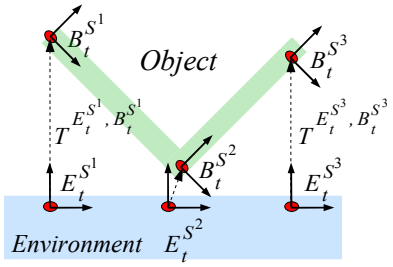

Fig. 6. Co-ordinate frames can be attached to an arbitrary number of local shapes, and local experts can be learned for each of these frames, predicting a distribution of how the frame may move next, given where it is at the present time step.

pose $E^{S_t^k}$ of a patch on a ground plane at the present time step, ignoring any information about the motions of the pushing finger. For the k-th such frame, we estimate *the local contact conditional density*:

$$p(T^{B^{S_t^k},B^{S_{t+1}^k}} | T^{E^{S_t^k},B^{S_t^k}}) \quad (10)$$

which represent probability density over possible rigid body transformations in the body frame of the k-th local contact. The subsequent motion of the object in the inertial frame can now be predicted as:

$$\max_{T_{in}^{B_t,B_{t+1}}} p_{local}((T^{I,B_t^l})^{-1} T_{in}^{B_t,B_{t+1}} T^{I,B_t^l} | T^{A_t,A_{t+1}}, T^{A_t^l,B_t^l}) \times p_{global}((T^{I,B_t})^{-1} T_{in}^{B_t,B_{t+1}} T^{I,B_t} | T^{A_t,A_{t+1}}, T^{A_t,B_t}, T^{B_t,O}) \times \prod_{k=1\dots N} p((T^{I,B^{S_t^k}})^{-1} T_{in}^{B_t,B_{t+1}} T^{I,B^{S_t^k}} | T^{E^{S_t^k},B^{S_t^k}}) \quad (11)$$

All joint and conditional densities are approximated by a variant of kernel density method with Gaussian kernels described in details in [2]. For simplicity, the density product 11 is maximised using the differential evolution optimization algorithm [6].

III. Simplified physics approach

A. Principle of minimum energy

The previous section presented a set of methods for learning to predict the behaviour of objects in simple robotic manipulation tasks. The methods incorporate information about objects' shapes and other physical properties in terms of distributions. The local distributions encode information about the behaviour of objects' local shape parts during interactions and can be shared among many objects. However, the global distribution is unique to a particular object or object category, therefore the generalization capabilities of such global distributions are limited, in particular with respect to objects of different shapes.

A simplified physics approach is an alternative method for predicting the motion of an object subjected to pushing action. The approach relies on the *principle of minimum energy* known from thermodynamics as a consequence the *second law of thermodynamics* applied to *closed systems*. The principle of minimum energy states that the total energy of a closed system decreases and reaches a local minimum value at equilibrium, where a closed system is a system with fixed entropy and other parameters such as volume or mass, but which can exchange energy with other connected systems [5].

A system consisting of a robot, an object and a ground plane can also be considered as a closed system. From the principle of minimum energy we know that the total energy of our system must reach a local minimum for a given amount of work introduced to the system. Each movement of a robotic finger, if it touches an object, produces some amount of work, which in the prediction scenario is unknown because the corresponding movement of the object is unknown. However, this movement can be computed by searching for such movements which minimize the produced amount of work, given known physical properties of the system.

A simplified physics approach uses a very simple model of physical interactions, which can be split into the physical phenomena and the corresponding work done by moving objects as follows:

1) *Mass* via work done by accelerating a given object.
2) *Gravity force* via work done while moving in a given potential field.
3) *Friction* via work done by two objects in contact moving in tangential direction. It is the simplest case of Coulomb's law of sliding friction with dynamic friction only.
4) *Restitution* via work done by two objects in contact moving in directions normal to the contacting surfaces.

B. Finding a trajectory at equilibrium

The simplified physics approach represents the object body by a set of N "volumetric" particles v_t^i with index i at discrete time step t randomly generated at time step $t = 0$ and then rigidly attached to the object throughout all prediction time steps (see Figure 7). Trajectory of an object is approximated by a sequence of rigid body transformations q which are found

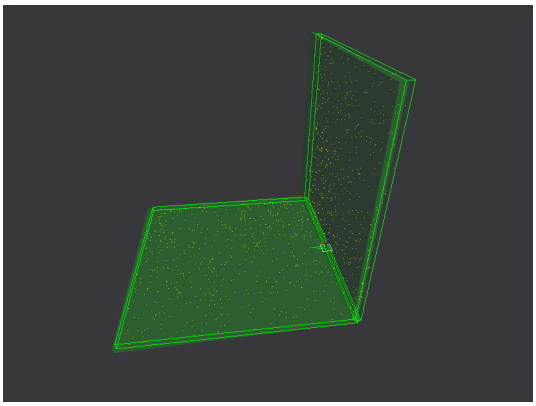

Fig. 7. A set of "volumetric" particles (yellow dots) representing the object body (green solid shape).

by solving a problem of minimizing the energy function $E(q)$ at each time step $t = 2, \ldots, T$:

$$\min_q E(q, t) \tag{12}$$

Energy function $E(q)$ consists of four work type-specific functions which correspond to four ways of producing work as described in the previous section:

$$E(q, t) = E^a(q, t) + E_i^g(q, t) + E_i^f(q, t) + E_i^r(q, t) \tag{13}$$

where each work function computes work during movement generated by q as follows[1]:

$$E^a(q, t) = C_a \| \sum_{i=1}^{N} (q v_{t-1}^i - 2 v_{t-1}^i + v_{t-2}^i) \| \tag{14}$$

$$E_i^g(q, t) = -C_g \sum_{i=1}^{N} G \cdot (q v_{t-1}^i - v_{t-1}^i) \tag{15}$$

$$E_i^f(q, t) = C_f \sum_{i \in V_f} \| q v_{t-1}^i - v_{t-1}^i \| \tag{16}$$

$$E_i^r(q, t) = C_r \sum_{i \in V_r} d_i(q v_{t-1}^i) \tag{17}$$

where $C_* \in \mathbb{R}^+$ are work type-specific constants, $G \in \mathbb{R}^3$ is the gravity vector, V_f is an index set of all particles which are in contact with the ground plane, V_r is an index set of all particles which penetrate a robotic finger or the ground plane with the corresponding penetration depth d_i.

Transformation q which minimizes $E(q)$ can be computed using e.g. a differential evolution optimization algorithm [6].

C. Probability density over trajectories

Energies $E(q)$ can be transformed into a probability density function over possible transformations q by using a Boltzmann distribution [5]:

[1]Work functions are only a crude approximation of real physical phenomena and do not even preserve physical units.

$$p_{Boltzmann}(E(q)) = \frac{\exp\left(-\frac{E(q)}{kT}\right)}{Z(T)} \tag{18}$$

where k is Boltzmann constant and T is temperature. $Z(T)$ is a partition function (a normalization constant) which for a given temperature can be computed from:

$$Z(T) = \sum_{q} \exp\left(-\frac{E(q)}{kT}\right) \tag{19}$$

Because a basic prediction scenario requires computation of only the most likely trajectory, normalization constant $Z(T)$ need not to be estimated and can be assumed a non-zero constant.

$p_{Boltzmann}(E(q))$ can be used as an approximation of the global conditional density function given by Equation 5 and it can be combined in a product with other experts as was discussed in the previous section. The global conditional density function can be replaced with $p_{Boltzmann}(E(q))$ so that Equation 11 now becomes:

$$\begin{aligned}\max_{T_{in}^{B_t,B_{t+1}}} & p_{Boltzmann}(E(T_{in}^{B_t,B_{t+1}}))\times \\ & p_{local}((T^{I,B_t^l})^{-1}T_{in}^{B_t,B_{t+1}}T^{I,B_t^l}|T^{A_t,A_{t+1}}, T^{A_t^l,B_t^l})\times \\ & \prod_{k=1\ldots N} p((T^{I,B^{S_t^k}})^{-1}T_{in}^{B_t,B_{t+1}}T^{I,B^{S_t^k}}|T^{E^{S_t^k},B^{S_t^k}})\end{aligned} \tag{20}$$

where symbol T stands for a rigid body transformation. The predicted object motion is a transformation $T_{in}^{B_t,B_{t+1}}$ which maximises the value of the above product.

$p_{Boltzmann}(E(q))$ depends on several constants which have to be estimated for a particular system, but crucially it also depends on temperature T. When temperature $T \to \infty$, $p_{Boltzmann}(E(q)) \to 1$ for any transformation q, consequently $p_{Boltzmann}(E(q))$ has no influence on a result of the maximization procedure 20. On the other hand, when temperature $T \to 0$, $p_{Boltzmann}(E(q))$ becomes very rugged, likely with a single peak only, so that the other factors in the product 20 have no impact on the maximization result.

IV. Results

A. Experimental setup

We have tested the introduced prediction algorithms in simulation experiments using PhysX physics engine [7], and in real experiments using 5-axis Katana robotic manipulator [8] equipped with a single rigid finger. We capture the motion of an object using a vision tracking system [9].

Multiple experimental trials were performed, in which a robotic arm equipped with a finger performs a random pushing movement of length approximately 25 cm towards an object placed at a random initial pose (Figure 8). In each experiment data samples are stored over a series of such random trials. Each trial lasts 10 seconds, while data samples are stored every $1/15$th of a second.

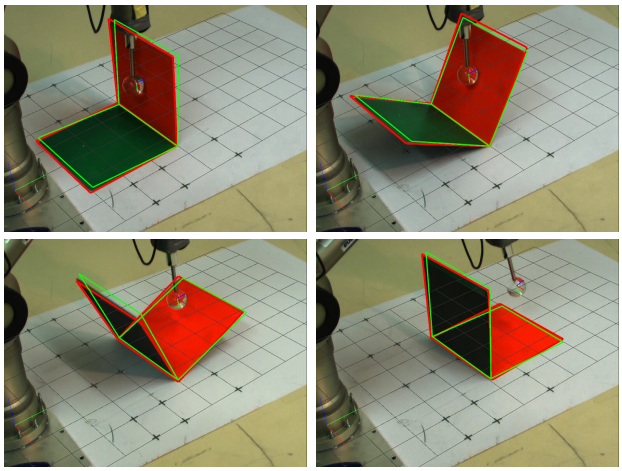

Fig. 8. A 5-DOF robotic arm equipped with a finger performs forward movements towards an object. Object behaviour varies depending on the initial object pose and finger trajectory. An example image sequence shows toppling behaviour. Orange wire frame denotes output of the vision based tracking system. Green wire frame shows predictions made by a simplified physics - note that the entire motion sequence is predicted before the physical push is initiated, without any recursive correction from visual feedback during the push execution.

B. Performance measure

In all experiments, we take the output of the tracked pose of a real object to be ground-truth, and compare it against predictions forecast by the simplified physics approach (Section III) or by the learned approaches (Section II). Prediction performance is evaluated as follows.

At any particular time step, t, a large number, N, of randomly chosen points $p_n^{1,t}$, where $n = 1 \ldots N$, are rigidly attached to an object at the ground-truth pose, and the corresponding points $p_n^{2,t}$ to an object at the predicted pose. At time step t, an average error E_t can now be defined as the mean of displacements between points on the object at the predicted pose and points on the object at the ground-truth pose:

$$E_t = \frac{1}{N}\sum_{n=1\ldots N} |p_n^{2,t} - p_n^{1,t}| \tag{21}$$

Note that for each robotic push action, we predict approximately 150 consecutive steps into the future, with no recursive filtering or corrector steps, hence it is expected that errors will grow with range from the initial object pose. We therefore find it more meaningful to normalize all errors with respect to an "average range", R_t, of the object from its starting position, defined as:

$$R_t = \frac{1}{N}\sum_{n=1\ldots N} |p_n^{1,t} - p_n^{1,0}| \tag{22}$$

For a test data set, consisting of K robotic pushes, each of which breaks down into many consecutive predictions over T time steps, we can now define an *normalized average error*:

$$E_{av} = \frac{1}{K}\sum_{k=1\ldots K}\frac{1}{T}\sum_{t=1\ldots T}\frac{E_t}{R_t} \tag{23}$$

For each set of test data, we also report an *normalized final error*, E_f which represents the typical discrepancy between prediction and ground truth that has accumulated by the end of each full robotic push:

$$E_f = \frac{1}{K} \sum_{k=1...K} \frac{|p_n^{2,T} - p_n^{1,T}|}{R_T} \tag{24}$$

We performed 10-fold cross-validation where at the beginning of each experiment all the trials are randomly partitioned into 10 subsets. Prediction was then subsequently performed (10 times) on each single subset, while learning (only for learned approaches) was always performed on the remaining 9 subsets of these trials. All the results were then averaged to produce a single estimation.

C. Performance of a simplified physics approach

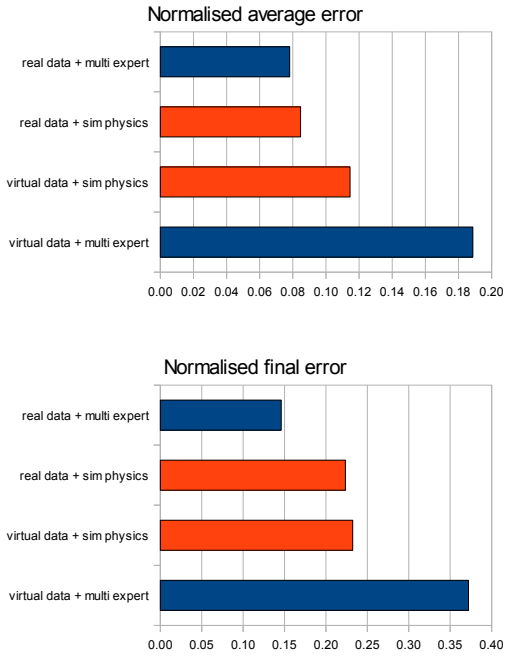

Fig. 9. If only simulation data is available for training of experts, then incorporating a simple physics predictor improves performance, approaching that of the ideal situation in which experts are trained on real observations of real objects. These charts compare the performance of a combination of learned experts which have been trained only on erroneous synthetic data, with and without the incorporation of an additional expert based on simple physics. In each case, performance has been assessed by then attempting to make predictions about motions of real objects which were not seen during training. For comparison, we also show data for the ideal situation in which experts have been exposed to examples of the real object during training.

Here we are interested in seeing how well our prediction systems can do, when no examples of captured real object motion are available for training, and instead we must rely on synthetic training data generated by simulation environments which do not correspond well with the real world. In all experiments, the various prediction approaches are tested by trying to make predictions about real objects being pushed by a 5-axis robot arm, and the predicted motions are compared against those captured by a vision system. We first train a combination of learned experts on synthetic push sequences, and then test by trying to predict the real motions of real objects being pushed by the robot. The resulting errors are shown in the bottom bar (virtual data + multi expert) in the charts of Figure 9. We next show that, by replacing one of the learned experts (the global or "gross body motion" expert) with an untrained simple physics predictor, we can significantly improve the predictions made about real objects. This result is represented by the second bar from the bottom in each chart (virtual data + sim physics). For comparison, we show the ideal situation (real data + multi expert) where a combination of learned experts has been trained on a large number of observed trajectories of real objects subject to real robot pushes.

If the reader compares the top bars of Figure 9 (real data + multi expert) against the second from top (real data + sim physics), it will be noted that simplified physics *does not perform as well* as a combination of purely learned predictors in cases where plenty of real-world observations are available for training. The advantage of the simplified physics contribution is that, in situations where prior experience of a real object is limited (e.g. when a robot encounters a new object), then a simplified physics contribution can improve on the performance of learned predictors that are merely trained on synthesized data from erroneous physics simulators.

Figure 8 and 10 shows some examples of successful predictions made by simplified physics. The toppling behaviour from Figure 8 is also correctly predicted by NVIDIA PhysX [7] game physics simulator. However PhysX struggles to provide correct predictions of sliding motion which involves large amount of rotation as it is visible in Figure 11. Similar rotational movements are reasonably well predicted by our simplified physics approach as it is shown in Figure 10.

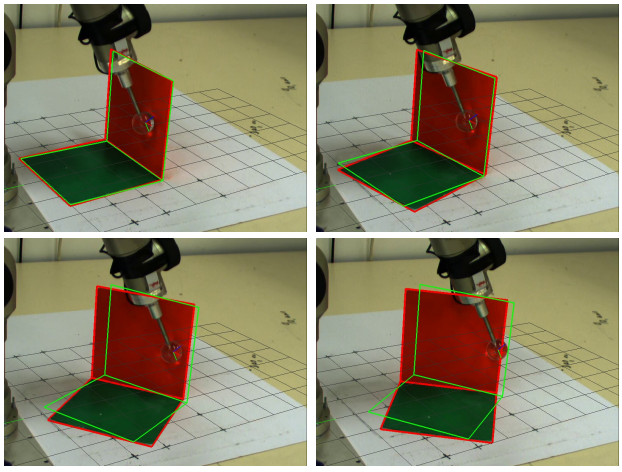

Fig. 10. An example image sequence shows sliding behaviour with large amount of rotational movement. Orange wire frame denotes output of the vision based tracking system. Green wire frame shows predictions made by simplified physics predictor.

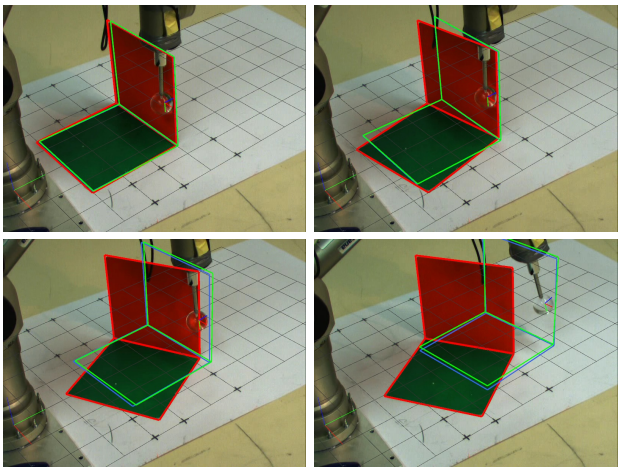

Fig. 11. An example image sequence shows sliding behaviour with large amount of rotational movement. Orange wire frame denotes output of the vision based tracking system. Green wire frame shows erroneous predictions made by predictor trained on virtual data provided by NVIDIA PhysX.

V. Conclusions

We have developed a number of methods for predicting the motions of manipulated rigid bodies, and we have also developed ways of combining these methods as a product of experts. Conventional physics simulators are often inadequate for making useful predictions about the interactions of real objects, and in many cases we find that learned predictors, trained on multiple example motions, perform much better. Unfortunately, it is not possible to use learned predictors for objects for which no prior training data is available, for example when a robot encounters a new object that it has not seen before. In such circumstances, one could attempt to train predictors on synthetic data, generated by a physics simulator, but the performance will be poor because such data is a poor representation of reality. this paper has shown that, in such situations, substantial advantage can be gained by incorporating (into the combination of experts) an expert that is not trained, but which infers the likelihoods of workpiece motions by applying the simple physics principle of minimum energy.

A useful property of the simple physics predictor is that it does not merely predict a single future object pose. Instead, by expressing the minimum energy principle in terms of a Boltzmann distribution, it is possible to predict an entire probability distribution over the space of possible candidate object motions. This is useful because it enables the opinions of the simple physics predictor to be combined with those of learned predictors via a simple product of densities approach. It is also useful in that, by supplying probabilities for candidate object poses, this can be used in predictor-corrector type recursive estimation problems such as vision-based tracking of manipulated objects using particle filters.

In the present work, predictions are made in advance for an entire 10 second push sequence, before the push is made, without any corrector or update steps from sensory inputs or recursive filtering. Ongoing work is exploring the use of these predictions as part of a predictor-corrector recursive estimation system for online visual tracking of manipulated objects. Since visual tracking data is necessary for training learned predictors, and the learned predictors may be useful for enhancing tracking, a bootstrapping problem is suggested for which the simple physics approach of this paper may prove a useful ingredient - the simple physics predictor may be used to enhance tracking, until sufficient data can be acquired to train a superior set of learned predictors.

VI. Acknowledgments

We gratefully acknowledge support for this research from EU-FP7-IST grants Nos 248273 (GeRT) and 215181 (CogX).

References

[1] M. T. Mason, *Mechanics of robotic manipulation*. MIT press, 2001.
[2] M. Kopicki, J. Wyatt, and R. Stolkin, "Prediction learning in robotic pushing manipulation," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–6, 2009.
[3] M. Kopicki, *Prediction learning in robotic manipulation*. PhD thesis, University of Birmingham, 2010.
[4] D. J. Duff, J. Wyatt, and R. Stolkin, "Motion estimation using physical simulation," in *IEEE International Conference on Robotics and Automation*, (Alaska), IEEE, May 2010. To be published.
[5] L. D. Landau and E. M. Lifshitz, *Statistical Physics, Part 1*, vol. 5 of *Course of Theoretical Physics*. Elsevier Butterworth-Heinemann, third ed., 1980.
[6] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
[7] N. PhysX, "Physics simulation for developers." http://developer.nvidia.com/object/physx.html, 2009.
[8] Neuronics AG, "Katana user manual and technical description." http://www.neuronics.ch, 2004.
[9] T. Mörwald, M. Zillich, and M. Vincze, "Edge tracking of textured objects with a recursive particle filter," in *Proceedings of the Graphicon 2009*, (Moscow, Russia), 2009.