Estimating part pose statistics with application to industrial parts feeding and shape design: new metrics, algorithms, simulation experiments and datasets

Péter L. Várkonyi

Abstract—**Part feeders take an unsorted bulk of identical parts and output them in a uniform orientation. Common feeders sort out those items that settle initially on one specific face, and further reorient them as desired. Quick estimators of the probability of settling on a given face facilitate the design of parts for efficient feeding and of the feeding lines themselves. Nevertheless the evaluation and the development of such estimators have been hindered by the lack of data. Here, I create and analyze a large, simulated dataset; evaluate estimators available in the literature by comparing their predictions to simulation results with the help of a custom-made metric; and propose new estimation algorithms. The new estimators offer viable alternative to the direct dynamic simulation of parts due to their low average errors.**

Note to practitioners—**Automated industrial assembly systems often take streams of unoriented parts as input, which are oriented by feeders. Many of the feeders reject all parts but those which rest on one desired face after falling into the feeder. The probability that a part settles on the right face has a dramatic effect on the efficiency of the feeder. In the paper I create a dataset on the behavior of irregularly-shaped parts by computer simulation. I compare the dataset to the predictions of simple estimators of these probabilities. Finally, I design a new estimator, which enables the quick and reliable prediction of the probabilities without the need to perform repeated experiments or computer simulations with every individual part. The new estimator is useful in those situations where other methods are too slow. For example, it can supply the user of computer aided design software with continuous, feedback on the 'feedability' of the part being designed.**

Index Terms—**assembly systems, estimation, part feeding, pose statistics, simulation**

1. Introduction

A. *Part feeding*

Parts used in automated assembly lines are often available in bulk and are oriented by feeders before entering the assembly line. Typically, the parts are dropped to a horizontal surface (e.g. a tray, a vibratory bowl or a conveyor belt) and settle on one of their faces. I refer to this initial phase of the orientation as 'pose selection'. After coming to rest, they are further processed by active manipulators (e.g. vibrated trays with depressions [1]; grippers [2], [3]; distributed manipulators [4], [5], [6], [7]; actively tilted trays [8]; throwing devices [9]) or passive physical barriers (fences [10], [11], [12], [13]; traps [14], [15]; pins [16]; 'blades' [17]). Some of these devices change the orientations of parts, whereas others are designed to sort badly oriented ones for recycling.

Although the core mechanism of a feeder is reused across different parts, the sorting and orienting devices require part specific design. Traditional methods of designing an efficient feeding line for a new part are slow and expensive. The main goals of researchers during the past two decades have been to facilitate feeder design by algorithmic planning and optimization [10], [11], [12], [14], [18] [19]; and to develop flexible or universal orienting devices [4], [5], [6], [7].

The majority of the orienting devices can only rotate parts about a vertical axis [20], but they cannot change their initial poses. Hence, the automated design of efficient feeders requires simple and reliable methods to predict (or manipulate [21]) the probabilities of settling in individual stable poses of the parts (i.e. pose statistics). The aim of the present paper is to improve our understanding of this aspect of part feeding.

B. *Part pose statistics*

There are two straightforward approaches to obtain the pose statistics of an individual shape: repeated physical experiments [22], and direct computer simulations [5], [23]. Experiments require the part to be manufactured as well as costly or tedious data processing. Computer simulation is much simpler, but generating a statistically reliable amount of pose statistics data by simulation is too slow for certain applications. Fast methods can be applied in computer-aided design software to provide the user with real-time feedback about the list of

The author thanks Tamás Krenn for writing parts of the simulation code, and anonymous reviewers for their valuable insights and recommendations. This work is supported by the Hungarian Scientific Research Fund under grant 104501 and by a Bolyai Research Fellowship.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. This includes a multimedia AVI format movie clip showing the simulated dynamics of a polyhedron; a data sheet of simulation results and predictions of estimators; source codes; and readme files. This material is 28 MB in size

The author is with the Department of Mechanics, Materials, and Structures, Budapest University of Technology, 1111 Budapest, Hungary (e-mail: vpeter@mit.bme.hu).

stable poses and the associated resting probabilities, while a part is being designed; they also facilitate the automated optimization of part shape for 'feedability' through a large number of iterative steps. This demand inspired the development of simple estimators [24], [25], [26], [27], [28], [29], [30], [31], which provide approximate pose statistics with low computational cost.

The downside of the estimators is their unknown reliability. Several studies have compared their predictions with experiments or simulations [24], [27], [25], [32], [33] and relatively low error rates have been reported. However, none of them has been benchmarked against more than a few specific objects or a restrictive class of shapes (e.g. square prisms). In this paper, I attempt to fill this gap by

1. producing a benchmark dataset of pose statistics via the numerical simulation of random polyhedra (see Sec. 3, and the downloadable material)
2. testing 5 existing estimators against data (Sec. 4.A-B), and
3. proposing 3 improved estimators (Sec. 4.C-E).

Next, I discuss the challenge of obtaining a high-quality benchmark dataset.

C. *Benchmarking the estimators*

Comparing and evaluating estimators requires a benchmark dataset. Both simulated and experimental datasets should be treated with caution because the difficulty of modeling impacts and friction [23]. The trajectory of an object falling onto a surface includes many impacts and the outcome of every single one depends on object shapes, sizes, materials, surface types and initial conditions in a highly nontrivial way. Ideally, an experimental benchmark dataset should cover all possible combinations of these factors. At the same time, all these details should be known about the system for which the estimation is made. Similarly, the impact and friction models embedded in a simulation code should reflect all these factors. These requirements are unrealistic, i.e. from the point of view of accuracy, neither experiments nor simulations are perfect.

Despite the sensitivity of an object's trajectory to the outcome of individual impacts, several earlier studies suggest that pose statistics are relatively robust against such details. The deviation of simulated statistics from experiments for a small set of objects is below 3% [25]; the effect of the quality of the support surface is also moderate (see [24], [27], [32] and Fig. 5). An attempt to rigorously prove the robustness of pose statistics is beyond the scope of the paper, but more evidence for this observation is presented in Sec. 5.

Motivated by the possibility to examine a large variety of object shapes, I use a simulated dataset as reference. My approach is supported by the above discussed robustness of pose statistics. The model is made as simple as possible, e.g. interactions between the objects and the underlying surface are modeled as perfectly frictionless.

2. Concepts and notations

This section is devoted to the description of technical concepts used in the paper.

A. *Pose*

An object's pose is a set of configurations invariant under rotations about a vertical axis, as well as under translations. For convex objects, configurations with the same point of the object's surface vertically under its center of mass correspond to the same pose. By considering a reference configuration of the object and by projecting its surface centrally to a unit sphere S_O about its center of mass O, one obtains a one-to-one correspondence between the poses and the points of the sphere (Fig. 1, [34]). For example, the 'north pole' of the sphere of poses corresponds to the pose, which is obtained by turning the reference configuration upside down. Accordingly, poses are also identified by three-dimensional unit vectors (**u**).

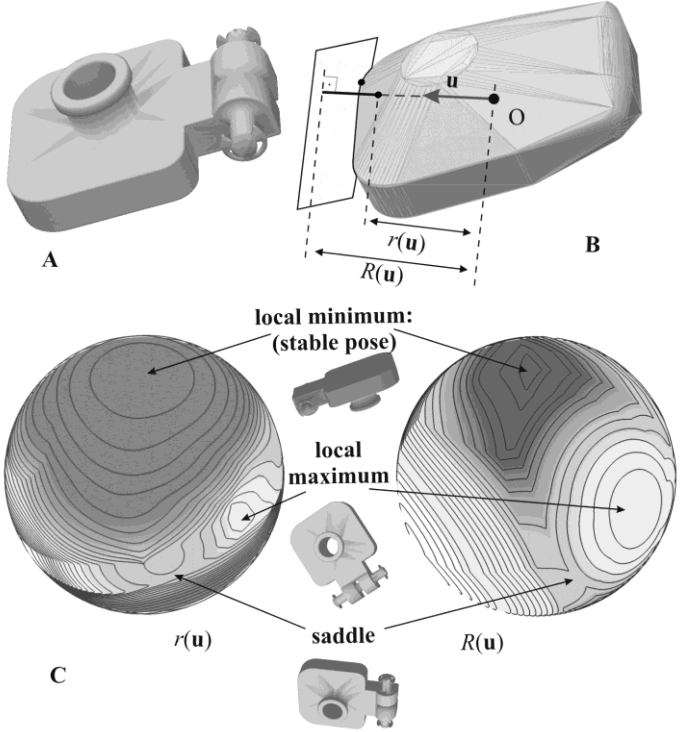

Fig. 1 A: example of a concave part (the body of a clamp) B: polyhedral approximation of its convex hull with definitions of the functions $r(\mathbf{u})$ and $R(\mathbf{u})$. The light grey plane touches the surface of the object and it is perpendicular to **u**. C: filled contour plots of the $r(\mathbf{u})$ (left) and the $R(\mathbf{u})$ (right) functions associated with the convex hull. Light colors indicate high values. Three poses of the object corresponding to a local maximum, a local minimum and a saddle point of the functions r and R are also displayed.

B. *Polyhedra*

The surface of any object can be approximated by a polyhedron (Fig. 1). A concave object can be replaced by its convex hull, and a non-triangular face can be divided to triangles. Thus, I only discuss convex triangular polyhedra. A polyhedron in a stable pose always rests on one of its faces (however not all faces are stable). To find stable faces, one can project the center of mass normally to the plane of each triangular face. A projection inside the triangle indicates stability.

It is worth pointing out that the approximation of a curved surface by a polyhedron may change the number of equilibrium poses. For example, a single equilibrium pose of the smooth object may turn into a small group of equilibrium poses including 'fake' stable poses [34]. The refinement of the polyhedral approximation reduces the distances between the equilibrium poses in such a cluster, but they do not necessarily disappear. One can deal with clusters of nearby stable poses by treating their members as identical, i.e. by summing the corresponding estimated probabilities. An extreme case of this side-effect occurs for parts containing a piece of spherical surface centered exactly at the part's center of mass. The polyhedral approximation replaces infinitely many neutral equilibria by a large but finite set of stable and unstable poses.

C. Functions describing the shape of an object

Given an object in a reference configuration with O in the origin, and an arbitrary three-dimensional unit vector **u**, there exists exactly one positive scalar $r(\mathbf{u})$ such that $r(\mathbf{u})\mathbf{u}$ belongs to the surface of the object. The function $r(\mathbf{u})$ can be used to describe the shape of the object. Another positive function $R(\mathbf{u})$ is defined as follows: $R(\mathbf{u})$ is the only positive number such that $R(\mathbf{u})\mathbf{u}$ belongs to a plane touching the object and perpendicular to **u** (Fig. 1). If the object is in pose **u** and it touches a horizontal plane from above, its potential energy is $mgR(\mathbf{u})$ where m is the mass of the object and g is the constant of gravity. Hence, a vanishing gradient of $R(\mathbf{u})$ corresponds to an equilibrium and local minima of R to stable poses. It can be shown that the gradients of $r(\mathbf{u})$ and $R(\mathbf{u})$ vanish at the same points; furthermore $r(\mathbf{u})\leq R(\mathbf{u})$ with equality at equilibrium poses.

D. Centroid solid angles

The solid angle of a surface embedded in 3D space from O (centroid solid angle) is the area of the central projection of the surface to the previously defined sphere S_O. There are closed formulas for the centroid solid angle of a spatial triangle [36] and for that of a circular sector with center C, provided that the plane of the sector is perpendicular to OC [37]. These formulas are used to calculate the centroid solid angle of

1. a face of a polyhedron (CSAE estimator, Sec. 4.B)
2. regions of S_O bounded by level curves of r or R (EAE, MEAE estimators, Sec. 4.D). The functions r and R are piecewise smooth; their level curves are composed of circular arcs as illustrated by Fig. 1.

3. Dynamic simulation

The main results of the paper are derived from a dataset on pose statistics generated by numerical simulation. n=100 drop tests have been simulated (Sec. 3.B,C) with each of 1057 random polyhedra (Sec. 3.A). The restitution coefficient of the impact model (Sec. 3.D) was first set to ρ=0.2. Then, the same was repeated with ρ=0.5 and 0.8. The simulations yield 3 datasets each consisting of N simulated probabilities, where N=6665 is the total number of stable poses summed for the 1057 objects. The probabilities are collected in three vectors: $\mathbf{s}_{0.2}=[s_{1,0.2},\ s_{2,0.2},\ \ldots,\ s_{N,0.2}]$, $\mathbf{s}_{0.5}=[s_{1,0.5},\ s_{2,0.5},\ \ldots,\ s_{N,0.5}]$ and $\mathbf{s}_{0.8}=[s_{1,0.8},\ s_{2,0.8},\ \ldots,\ s_{N,0.8}]$, whereas the vector of all 3N probabilities is denoted by **s**. The comparison of $\mathbf{s}_{0.2}$ and $\mathbf{s}_{0.8}$ sheds light on the effect of parameter ρ on pose statistics (Sec. 5.A), and the whole dataset **s** is used to evaluate and to fit the estimators of pose statistics (Sec. 4).

The polyhedra of the dataset have 2 to 21 stable poses and the associated probabilities are in the range 0-0.75. There is no object with a unique stable pose in the dataset.

Details of the object generation and the dynamic simulation are given below. The algorithm has been implemented under MATLAB R2010b. The run time of a single drop test on a PC (CPU: Intel Core2 2.5 GHz; OS: Microsoft Windows 7) is in the range of 0.2-5 sec. The execution time is low if the coefficient of restitution for impacts is low or if the stable facets are large. A video illustrating the simulation, a table of simulation results, and program codes are available as multimedia files.

A. Random shapes

Convex, random polyhedra are generated by the following algorithm:

- Random points are generated in 3-space. Each coordinate of each point is drawn from a normal distribution with mean 0 and variance 1; the number of points is 4+ξ, where ξ is a random number from a geometric distribution with parameter 0.1.
- The convex hull of the points (found by any standard method) provides the surface of the object. This is a convex polyhedron with triangular faces. In the actual dataset the number of vertices is in the range 4-23; the number of faces is between 4 and 42. The mass, the location of the center of mass, and the moment of inertia tensor of the object are calculated by assuming constant unit mass-density inside the polyhedron.
- The object is resized to set the maximum of the $r(\mathbf{u})$ function to 1.
- To include objects representing inhomogeneous and/or concave parts, the center of mass is shifted by random numbers d_i (i=1,2,3) along the eigenvectors of the moment of inertia tensor. The variables d_i are drawn from uniform distributions over the intervals $(-q_i,q_i)$ where q_i are the corresponding radii of gyration of the object (i.e. q_i^2 are the eigenvalues of the moment of inertia tensor divided by the mass of the object).

B. State variables of the objects and initial conditions

The motion of the objects is described in a Cartesian reference frame with axes x-y-z. The z axis points vertically upwards. The support surface is the z=0 horizontal plane. The dynamic state of a rigid body is determined by the vertical coordinate $z_O(t)$ of its centre of mass and its time derivative $\dot{z}_O(t)$; a 3 by 3 rotation matrix **R** representing its orientation; and its 3 by 1 angular velocity vector **ω**. The horizontal dynamics of O is trivial because all external forces acting on the object are vertical. Initially (t=0), the object is high above the surface relative to its size ($z_O(0)$=20). The initial orientation is chosen randomly with uniform distribution using an algorithm from [38]; the initial velocity and angular velocity are zero.

C. Continuous dynamics

The moving polyhedron is in one of the following contact modes: free fall, sliding on a vertex, sliding on an edge (i.e. 2 vertices), and sliding or resting on a face (i.e. 3 vertices). Accordingly, it is assumed that the ground exerts nonnegative, vertical contact forces to k=0, 1, 2 or 3 vertices of the object, which are in contact with the ground. More than 3 contact points are also possible if an object has coplanar triangular faces, but this scenario does not occur for shapes obtained by taking the convex hull of random points. The Newton-Euler equations and k contact constraints are used to determine the k contact forces together with the resulting acceleration and angular acceleration of the object. This way, the continuous motion of the object is determined by integrating a system of first order, ordinary differential equation in the variables z_O, $\dot{z}_O = v_O$, $\mathbf{R}$, $\boldsymbol{\omega}$ (see Appendix A).

The integration of the ODE is interrupted whenever an impact between the object and the surface is detected, or whenever one of the contact forces becomes negative.

D. *Impacts and Zeno points*

Single-vertex impacts are assumed to be frictionless (i.e. impact momenta are vertical) with Newtonian coefficient of restitution ρ. The impact impulse, and the post-impact velocity of the object are determined uniquely by the linear and angular momentum laws and the coefficient of restitution. Simultaneous impacts at several vertices are not determined uniquely by these laws. I treat them as sequences of single-vertex impacts. In each step, the vertex with the highest velocity pointing towards the underlying surface is assumed to undergo an impact. The modelling of simultaneous impacts is not possible without ad hoc rules such as the one described above [39]. It is assumed that pose statistics are not sensitive to this type of modelling error (see Sec. 1.C).

Interacting rigid bodies establish permanent contacts via special events called ‘Zeno points’. Zeno points are accumulation points of infinitely many impacts of exponentially decreasing intensity [36]. For example, a partially elastic rigid ball bouncing vertically on a horizontal support plane comes to rest through a Zeno point. A naive simulation in which each impact is treated as an individual event can never pass a Zeno point [41]. To avoid this pitfall, the simulation code includes Zeno point detection (Appendix B).

E. *Termination*

The simulation is completed if a face of the object establishes contact with the surface. The object conserves its 0 initial angular momentum about the z axis, hence face- contact implies immobility.

4. Estimators of pose statistics

A. *Measuring the quality of an estimator*

Let σ denote the root mean square deviation of two datasets of equal size N:

$$\sigma^2(\alpha,\beta) \overset{def}{=} N^{-1}\sum_{j=1}^{N}\left(\alpha_j-\beta_j\right)^2 \qquad (1)$$

To evaluate the estimators, the set **s** of simulated probabilities is compared with the predictions **e**=[e_1, e_2,…e_{3N}] of an estimators and with the unknown exact probabilities **p**=[p_1, p_2,…p_{3N}]. The deviation of **e** and **s** is caused partly by random noise and partly by the imperfectness of the estimator. The two components can be partitioned:

$$\begin{aligned}\sigma^2(\mathbf{e},\mathbf{s}) &= (3N)^{-1}\sum_{j=1}^{3N}\left(e_j-s_j\right)^2\\ &= (3N)^{-1}\sum_{j=1}^{3N}\left(e_j-p_j+p_j-s_j\right)^2\\ &= \sigma^2(\mathbf{e},\mathbf{p})+\sigma^2(\mathbf{p},\mathbf{s})-\underbrace{\frac{2}{3}N^{-1}\sum_{j=1}^{3N}\left(e_j-p_j\right)\left(s_j-p_j\right)}_{\overset{def}{=}\,C}\end{aligned} \qquad (2)$$

where $\sigma(\mathbf{e},\mathbf{p})$ measures the error of the estimator; $\sigma(\mathbf{p},\mathbf{s})$ represents noise in the simulation results (due to the finiteness of n) and the last term C is the covariance of the two errors.

The simulated probabilities are obtained by averaging results from a finite number (n) of simulated drop tests for each object. Thus, they are random numbers drawn from binomial distributions with n trials and probability parameters p_1, p_2,…p_{3N}. (Some of these probabilities belong to different stable poses of the same object, i.e. they are dependent.) The value of $\sigma(\mathbf{p},\mathbf{s})$ can be estimated using the formula of the variance of binomially distributed random variables:

$$\sigma^2(\mathbf{p},\mathbf{s}) \approx \left(3N\right)^{-1}n^{-1}\sum_{j=1}^{3N}p_j(1-p_j) \qquad (3)$$

The equality marked by ‘≈’ is approximate due to the finiteness of N (though the error terms are neglected as N is much larger than n). As a further approximation, the unknown values p_j in (**3**) are replaced by the known values s_j, yielding $\sigma^2(\mathbf{p},\mathbf{s}) \approx 0.0327$.

The term C in (**2**) is much smaller than $\sigma^2(\mathbf{p},\mathbf{s})$ unless the estimator is overfitted to the noisy observations. (Extreme overfitting, i.e. e_j=s_j would yield C=2$\sigma^2(\mathbf{p},\mathbf{s})$.) Some estimators examined in this paper are based on physical intuition rather than curve fitting; others have a small set of parameters fitted to a relatively large set of data. Hence, overfitting can be excluded, and C is negligible. Thus the error of the estimator can be approximated by

$$\sigma(\mathbf{e},\mathbf{p}) \approx \left[\sigma^2(\mathbf{e},\mathbf{s}) - 0.0327^{\ 2}\right]^{1/2} \qquad (4)$$

This quantity is used to compare and to evaluate the estimators.

If **p** was known, a graphical illustration of the partitioned noise term $\sigma(\mathbf{p},\mathbf{s})$ could be obtained by drawing random numbers r_1, r_2,…r_{3N} from binomial distributions with n=100 trials and probability parameters p_1, p_2,…p_{3N}; and by plotting them against the probability parameters. Unfortunately, **p** is unknown, but one can use the approximations **s** as input instead. The (r_i, s_i) points are off-diagonal (Fig. 2) due to the finiteness of n. Later, several plots of estimated vs. simulated results will be shown (Fig. 4). The points in these plots are even more off-diagonal due to estimator errors. Fig. 2 can be interpreted as the plot of a ‘perfect estimator’, which predicts **p** without any error.

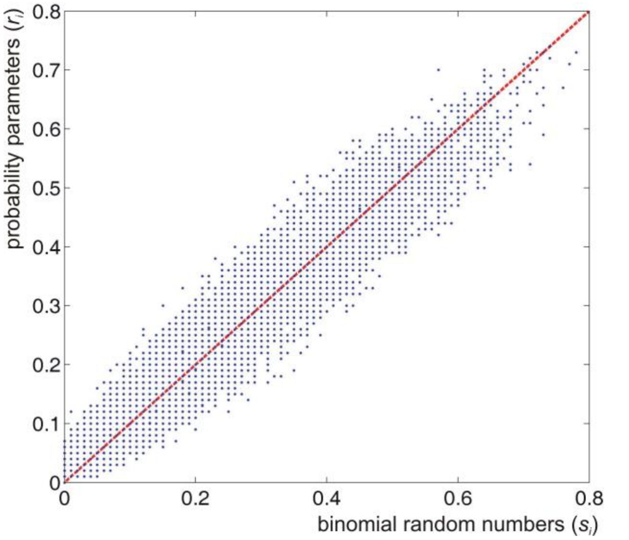

Fig. 2: Illustration of the noise caused by the finiteness of *n* (see main text for explanation)

B. Existing estimators

In one of the first papers on pose statistics, [24] proposed an estimator based on the extension and the depth of the potential energy valleys associated with the stable poses of an object (energy barrier method). The definition of the 'energy barrier' was given originally only for cylinders and square prisms. As it is not clear how to extend it to an arbitrary shape, this concept is not examined here.

Another early work on the topic [27] hypothesized that the probability of resting on a face is roughly proportional to the centroid solid angle Ω_i of the face from the center of mass. This is motivated by the fact that in a uniformly distributed random pose, the probability that the centroid is above face *i* would be $\Omega_i/4\pi$. This estimator assigns positive probability to unstable faces, which can be corrected by assuming that the probabilities associated with unstable faces are 0, and those of stable faces are proportional to their centroid solid angles:

$$e_i = \Omega_i / \sum_{stable\ \ faces} \Omega_j \qquad (5)$$

This is referred to as centroid solid angle estimator or CSAE. A diagram of the CSAE analogous to Fig. 2 is plotted in Fig. 4A. The latter point-cloud occupies a wide region around the main diagonal, which indicates a significant estimator error. The error of the CSEA calculated by (**4**) is 0.085. The surface area of faces can also be used as part pose estimator in a similar manner [24]. This estimator yields similar results to the CSAE with higher error (0.112).

References [28], [29], and other works of the same authors investigate the hypothesis that the probability of landing on a stable face is proportional to the ratio Ω_i/d_i where d_i is the distance of the face from the center of mass (angle-distance ratio estimator or ADRE). The estimated probabilities are obtained by a formula analogous to (**5**). The error of the ADRE is 0.078, i.e. it is slightly better than the CSAE (Fig. 4B).

In [26], the CSAE is complemented by a physically inspired algorithm to distribute the probabilities associated with unstable faces among stable ones. The CSAE is used in [26] to estimate the initial probability of landing on a given face as the object is dropped. If the initial face is unstable, the object is assumed to topple over one of its three edges to an adjacent face quasi-statically. The toppling continues until a stable face is reached. A simple physically inspired rule is chosen to determine, which way the polyhedron topples. This model of toppling motion can be represented by an acyclic directed 'toppling graph' (its vertices correspond to faces of the polyhedron, and its edges indicate the directions of toppling), on which the state of the object evolves. The initial probability distribution predicted by the CSAE is propagated along the graph towards stable states to obtain the quasi-static estimator (QSE, Fig. 4.C), which has a lower error (0.070) than the CSAE.

By examining the experimental pose statistics of a few objects, several authors have noticed that the previously described estimators tend to underestimate the probabilities associated with large faces and overestimate the likelihood of landing on smaller ones. This tendency – also reflected by the nonlinear trends of the point clouds in Fig. 4.A-C – was attributed by [25] to the fact that an object resting on a small face is much more sensitive to perturbations than one resting on a large face. To address this issue, the QSE was modified in the following way [25]: the initial probabilities (coming from CSAE) were redistributed between each pair of adjacent faces according to a simple phenomenological rule that increased the initial probability associated with large faces. The modified initial probabilities were propagated down the toppling graph (as in the case of the QSE) to obtain the predictions of the perturbed quasi-static estimator (PQSE). The PQSE was tested against experiments and simulations with three specific objects. It was found to outperform the QSE, indeed its performance was comparable to that of direct numerical simulations. When applied to my set of random objects, this estimator assigns negative initial probabilities to some small faces, which is meaningless. In such situations, I reduce the amount of redistribution to keep the initial probabilities nonnegative. The predictions of the PQSE (Fig. 4.D) lack the nonlinear trend of the CSAE however the estimator error remains roughly the same (0.072).

C. Improving the performance of estimators by nonlinear fitting

Inspired by the initial observations of [25], I examine a purely mathematical method to avoid the overestimation of probabilities associated with small faces. The results e_j of the QSE estimator are used as basis. The parametric nonlinear function

$$f(x) = x + x(1-x)\left(c_0 + c_1 x + c_2 x^2 + c_3 x^3\right) \qquad c_i \in \Re \qquad (6)$$

is fitted to the points of Fig. 4.C to minimize the variance $\sigma^2(f(\mathbf{e}), \mathbf{s})$. The fitted values are c_0=-0.874; c_1=8.03; c_2=-16.3; c_3=7.70. It is hypothesized that the probabilities of landing on various faces of a polyhedron are proportional to $f(e_j)$, leading to a formula similar to (**5**). The error of the new 'nonlinear quasi-static estimator' (NQSE) is significantly lower (0.062) than any of the previous ones (Fig. 4.E).

D. EAE: a new estimator

The philosophy behind the QSE estimator and its variants was to get rid of complex dynamics by considering a highly damped, quasi-static model of the moving object, and to derive estimations from the 'tamed' model. The new estimator presented here attempts to capture the statistical properties of the complex dynamics instead of neglecting it. The main idea of the estimator is illustrated by Fig. 3.

Let $E(t)=U(t)+K(t)$ denote the total mechanical energy, the potential energy, and the kinetic energy of the object. The reference level of the potential energy is the supporting surface. The object may not penetrate into the surface, yielding $U(t)\geq mgR(\mathbf{u}(t))$; at the same time, $K(t)$ is nonnegative. These inequalities imply

$$R(\mathbf{u}(t))\leq E(t) / (mg) \tag{7}$$

i.e. those poses for which $R(\mathbf{u})$ exceeds a certain threshold, are not reachable by the object. The initial mechanical energy of the object is enough to make every pose reachable. However its motion is accompanied by energy absorption, which is assumed to happen either continuously or through small steps. The set of reachable poses shrinks gradually (Fig. 3D-F). The topological changes of the reachable set have been investigated thoroughly in [34]. The next paragraph gives a brief description of these results. The function R has s local minima corresponding to the stable poses. Accordingly, there are s-1 energy levels at which the number of disconnected components of the reachable set increases by one. The splitting events occur at saddle points of the function $R(\mathbf{u})$ (but not necessarily all saddles are involved in splitting events, see Fig. 3D). The fragmentation of the reachable set can be described by a 'splitting graph' (Fig. 3C) in which rectangular nodes represent connected components of the reachable set at various energy levels. Each node is labelled by the list of stable poses contained by that component. Pairs of edges represent splitting events, and their positions show the corresponding energy levels. Those energy levels, at which a component disappears, are marked by circle-shaped nodes at the bottom of the graph; however these values are not used by the estimator.

The mechanical energy and the actual pose of the object together determine a node of the splitting graph. This node is referred to as the discrete state of the object. During its motion, the discrete state of the object changes from time to time. The *Energy absorption estimator* (EAE) is based on a phenomenological description of this process. A Markov chain is considered on the graph. The transition probabilities are assumed to depend on geometric properties of the examined object. Specifically, if a component Γ_i of the reachable set splits to two parts (Γ_{ia}, Γ_{ib}) at some energy level E^*, the transition probabilities π_{ia}, π_{ib} are assumed to be proportional to the centroid solid angle of the two components at energy level E^*:

$$\begin{aligned}\pi_{ia} &= \Omega(\Gamma_{ia})/\left(\Omega(\Gamma_{ia})+\Omega(\Gamma_{ib})\right)\\ \pi_{ib} &= \Omega(\Gamma_{ib})/\left(\Omega(\Gamma_{ia})+\Omega(\Gamma_{ib})\right)\end{aligned} \tag{8}$$

An algorithm to identify the connected components of the reachable set at the energy levels corresponding to saddles of $R(\mathbf{u})$ is outlined in [34]. The complexity of this algorithm is linear in the number of vertices of the polyhedron. In the current work, I use a different one, which is sketched in [42].

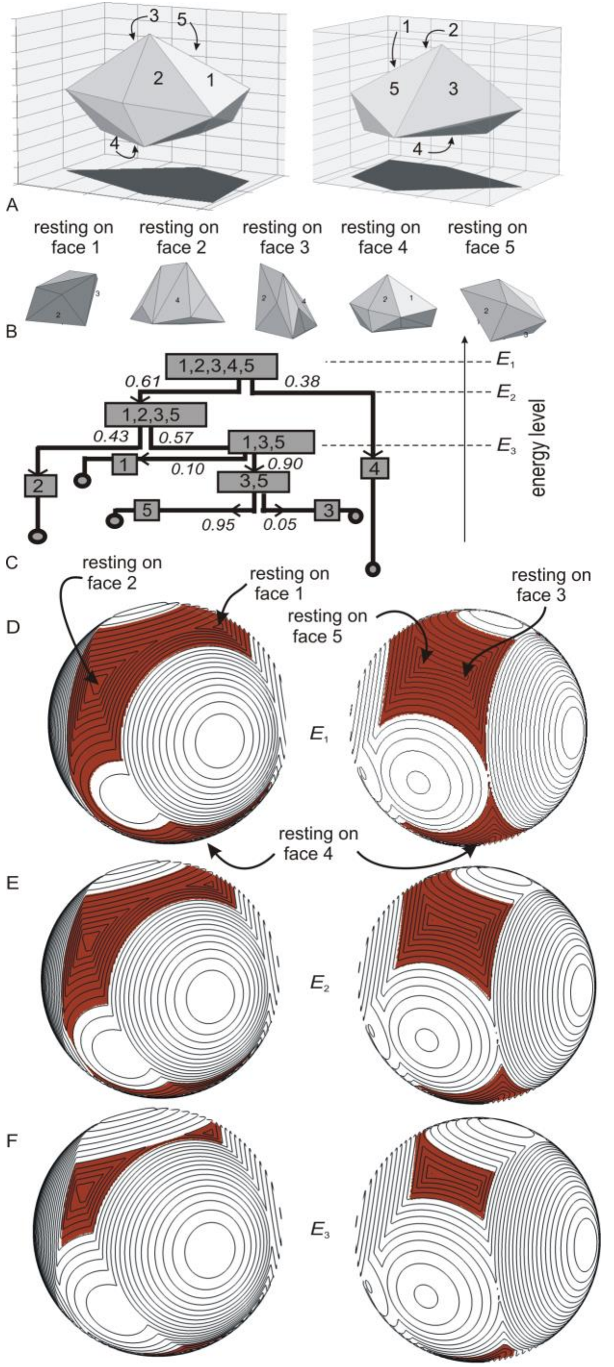

Fig. 3 A: front (left) and rear (right) view of one of the polyhedra used in the simulation. Resting on any of the 5 enumerated faces corresponds to a stable pose. B: the stable poses of the polyhedron C: the splitting graph of the polyhedron with transition probabilities. There are four critical energy levels, at which the reachable set splits. According to (7), the critical energy levels are those at which a saddle point of the function $R(\mathbf{u})$ leaves the reachable set. Circles represent the potential energy levels at which a component of the reachable set disappears. D-F: front (left) and rear (right) views of the sphere representing the space of poses, with level curves of the $R(\mathbf{u})$ function. The black areas are the set of reachable poses at three energy levels (dashed lines in panel C). E_1 is not critical because the reachable set remains connected, but E_2 and E_3 are critical.

Initially, the object's discrete state corresponds to the node at the top of the graph with probability 1. Eq. (8) is used to calculate the likelihood of reaching each other discrete state. Among these, the probabilities associated with the absorbing states at the bottom of the tree (each corresponding to one stable pose) are the predictions of the EAE estimator.

The error of the estimator is 0.068. In comparison with previous results, this error level is not very impressive. Nevertheless, much better results are obtained if the function *R* is replaced by *r* in (7). Even though the author has no physical argument why such a modification should be beneficial, the error of the *Modified energy absorption estimator* (MEAE) is only 0.040 (Fig. 4.F). The algorithm of the MEAE estimator is described in Appendix C.

The run time of a MATLAB-based implementation of the MEAE estimator is between 0.1 and 1.5 second depending on the complexity of the polyhedron. The run time of the EAE estimator is slightly higher. The code has not been optimized for speed.

The computational complexities of the estimation algorithms are O(*v*) where *v* is the number of vertices (Appendix B, [42]). This is lower than the O(*v*log*v*) complexity of finding the convex hull.

E. Combining several estimators

The NQSE and the MEAE estimators capture different aspect of the dynamics. It is likely that improved results can be obtained by combining them. Specifically, parameter *c* of the convex combination

$$\mathbf{e}_{combined} = c\mathbf{e}_{MEAE} + (1-c)\mathbf{e}_{NQSE} \qquad (9)$$

has been tuned to minimize $\sigma^2(\mathbf{e}_{combined}, \mathbf{s})$. The error of the combined estimator with *c*=0.75 shows further reduction (0.036). Linear and nonlinear combinations of more than 2 estimators have also been examined, however they did not yield significantly better results. This is probably caused by the strong correlation of the predictions of different estimators.

F. Outlier cases

Estimators produce exceptionally large errors in some cases. The maximum deviations of various estimators from results obtained by simulation are summarized in Table I. The best performance is that of the new combined estimator. I also examined the 3 largest observed deviations of the combined estimator: 32.0% (facet 3 of object 354 of the dataset, ρ=0.8); 31.9% (facet 6; object 413; ρ=0.2) and 29.5% (facet 7; object 431; ρ=0.5). The finiteness of *n* is partly responsible for these outlier cases. To separate this effect, 1000 drop tests (instead of 100) were simulated with these 3 particular objects. The errors of the estimator were reduced to 20.8%, 22.5% and 17.6%. This result suggests that the combined estimator never or extremely rarely produces errors significantly exceeding 20%.

5. VERIFICATION OF THE RESULTS

It is a crucial assumption of the paper that pose statistics are robust against modelling errors. The detailed discussion of this question is beyond the scope of the paper, but the plausibility of the assumption is illustrated by a systematic examination of the coefficient of restitution ρ and by a comparison of simulated results with experiments under various conditions.

A. Sensitivity to the coefficient of restitution

The deviation of the simulated datasets obtained with ρ=0.2 and ρ=0.8 is $\sigma(\mathbf{s}_{0.2}, \mathbf{s}_{0.8}) = 0.0728.$ This can be partitioned similarly to the techniques described in Sec. 4.A. The Biennaymé formula for the variance of sums of random variables; the connection between the raw moments and the central moments of a probability distribution; and the formula of the variance of a binomially distributed random variable yield

$$\begin{aligned}
&\sigma^2(\mathbf{s}_{0.2},\mathbf{s}_{0.8}) \approx \sigma^2(\mathbf{s}_{0.2},\mathbf{p}_{0.2}) + \sigma^2(\mathbf{p}_{0.2},\mathbf{s}_{0.8}) \\
&\approx \sigma^2(\mathbf{s}_{0.2},\mathbf{p}_{0.2}) + \sigma^2(\mathbf{p}_{0.2},\mathbf{p}_{0.8}) + \sigma^2(\mathbf{p}_{0.8},\mathbf{s}_{0.8}) \\
&\approx N^{-1}n^{-1}\sum_{j=1}^{N} p_{j,0.2}(1-p_{j,0.2}) + \sigma^2(\mathbf{p}_{0.2},\mathbf{p}_{0.8}) \qquad (10) \\
&+ N^{-1}n^{-1}\sum_{j=1}^{N} p_{j,0.8}(1-p_{j,0.8})
\end{aligned}$$

The unknown values $p_{j,\rho}$ in (10) are replaced by the known values $s_{j,\rho}$, which lets one estimate the deviation of the unknown 'exact' datasets as

$$\begin{aligned}
&\sigma(\mathbf{p}_{0.2},\mathbf{p}_{0.8}) \approx ... \\
&\left(\sigma^2(\mathbf{s}_{0.2},\mathbf{s}_{0.8}) - N^{-1}n^{-1}\sum_{j=1}^{N} s_{j,0.2}(1-s_{j,0.2}) - N^{-1}n^{-1}\sum_{j=1}^{N} s_{j,0.8}(1-s_{j,0.8})\right)^{1/2} \\
&= \left(0.0728^2 - 0.0321^2 - 0.0332^2\right)^{1/2} = 0.0563
\end{aligned}$$

Hence, the effect of the coefficient of restitution on pose statistics is moderate (on average: 5.6%) despite the fact that the trajectory of an individual falling object can fundamentally change its character in response to variations of ρ. This result confirms the robustness of pose statistics against modeling errors.

B. Comparison with experiments

The pose statistics of square prisms of various length-to-width ratio have been investigated experimentally among others by [27] and [32]. Both works report on drop tests with hard and soft support surfaces. In [32], the surface is vibrated. The results of these experiments are summarized in Fig. 5. The predictions of various estimators and simulation results are also shown.

As already noted by [27], the quality of the support surface has significant influence on the experimental results. Depending on the length/width ratio, the effect of the surface type on the results is between 0 and 15%. This is comparable to the effect of the coefficient of restitution found in Sec. 5.A.

The simulation results agree well with experiments obtained with a hard surface, but they deviate significantly from the soft surface results. This observation suggests that the results of the paper do not apply very well to objects dropped onto soft surfaces, and further work using a different benchmark dataset is desirable in that case.

TABLE I
COMPARISON OF ESTIMATORS.

	name	average error	largest observed deviation from simulation
existing estimators	CSAE	8.5%	43.5%
	ADRE	7.8%	40.9%
	QSE	7.0%	45.8%
	PQSE	7.2%	42.5%
new estimators	NQSE	6.2%	44.9%
	EAE	6.8%	58.0%
	MEAE	4.0%	32.7%
	combined	3.6%	32.0%

Finally, comparison of the estimators reveals a somewhat surprising result. The combined estimator shows the best fit to the benchmark dataset, but it has relatively large errors (up to 17% for some length/width ratios and surface types) in the case of square prisms. The ADRE and the EAE perform significantly better for this particular class of objects.

6. Conclusions

Systematic comparison and evaluation of existing estimators of pose statistics have been performed in the paper. It was found that the average errors of the estimators are between 7 and 11% (Table I). Such an error level can be tolerated in some situations, but estimations based on direct numeric simulation are significantly more accurate (though not perfectly accurate as discussed in Sec. 1.C). The increasing availability of cheap computation makes simple estimators less attractive, nevertheless there are applications for which dynamic simulation is too slow. For this reason, estimators with low error rates are of practical interest. In the current paper, a new estimators has been proposed with average error 3.6% and maximum error near 20% (based on a large, random dataset).

To measure the performance of the estimator, simulations with a frictionless model have been used as reference. The effect of modeling errors and the influence of experimental conditions on pose statistics have been tested in several ways. Large variations of the coefficient of restitution change the probabilities by 5.6% on average. According to earlier experiments with square prisms, the quality of the support surface has an effect of similar magnitude. On the one hand, these observations indicate that further significant reduction in estimator error is not possible without treating the coefficient of restitution and other parameters (e.g. friction coefficient) as known. This paper does not examine such extensions, because these parameters are often unknown in real-life situations. On the other hand, understanding the parameter-dependence of pose statistics requires further effort (e.g. the systematic variation of physical parameters in physical experiments or in simulations with a more detailed model).

Appendix

A. Equations of motion

The ODE describing the continuous motion of the object consists of the following parts [42]:

$$\text{Newton's law: } \dot{v}_O = -g + m^{-1}\sum_{j=1}^{k} N_j \qquad (12)$$

$$\text{Euler's rigid body eq.: } \dot{\boldsymbol{\omega}} = -\mathbf{I}^{-1}\left(\Omega\mathbf{I}\boldsymbol{\omega} + \sum_{j=1}^{k} N_j\mathbf{r}_j \times \mathbf{u}_z\right) \qquad (13)$$

$$\text{kinematic identity: } \dot{z}_O = v_O \qquad (14)$$

$$\text{kinematic identity: } \dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\Omega} \qquad (15)$$

where g is the constant of gravity; m is the mass of the object and $\mathbf{I}$ is its moment of inertia tensor in local coordinates; N_j are the contact forces, k is the number of sliding contacts, and $\mathbf{r}_j$ are the position vectors of the corresponding vertices in local coordinates; $\mathbf{u}_z$ is an upward pointing vertical unit vector. $\boldsymbol{\Omega}$ is the matrix corresponding to the $\boldsymbol{\omega}\times$ cross product operation (i.e. $\boldsymbol{\Omega}\mathbf{x}=\boldsymbol{\omega}\times\mathbf{x}$ for any $\mathbf{x}$); $\boldsymbol{\Omega}$ contains 0's and the elements of $\boldsymbol{\omega}$.

The contact forces N_j are determined from the non-penetration constraints $z_O+\mathbf{u}_z^T\mathbf{R}\mathbf{r}_j=0$. Substitution of (14), (15) into the second time derivative of this constraint yields

$$\dot{v}_O + \mathbf{u}_z^T\mathbf{R}\left(\boldsymbol{\Omega}\boldsymbol{\Omega} + \dot{\boldsymbol{\Omega}}\right)\mathbf{r}_j = 0 \qquad (16)$$

Substituting (**12**), (**13**) into (**16**) yields a system of linear algebraic equations for the contact forces.

B. Detecting Zeno points

An approaching Zeno point is indicated by an impact event, at which the pre-impact vertical velocities of all vertices that are very close to the ground (the threshold 10^{-9} is used in the simulation), are either positive (moving upwards) or very close to zero (threshold: -10^{-5}) and the sets of 'slowly moving vertices' is nonempty. If this situation is detected, I apply appropriately chosen instantaneous, vertical impulses at the set of slowly moving vertices, in order to change their vertical velocities to 0. The necessary impulses are determined uniquely by the linear and angular momentum laws. This impulse is an approximate replacement for the infinitely many small impacts leading to the Zeno point. After the impulse, the contact mode of the object switches to sliding on these vertices.

C. Algorithm of the MEAE estimator

The MEAE estimator performs the following steps:
1: the stable faces are found by testing every face one by one. For specific formulas, see [34] [42]. Those nodes of the splitting graph, which are labelled by a single stable pose are drawn.
2: the saddle points of r are found by testing every edge one by one. The edges with saddle points are $l_1,l_2,\ldots,l_H$ and the critical values of r are $r_1\leq r_2,\leq\ldots\leq r_H$.
3: The distance of every edge (including those without saddle points) from the center of mass is determined.
4: The following steps are repeated for every critical value r_i, i=1,2,…,H.
4A: a connectivity graph of faces is generated by testing every edge one by one. Two faces are connected if they share an edge, which is closer to the centre of mass than r_i.

4B: the connected components of the connectivity graph are found using a breadth-first traverse.

4C: if the two faces adjacent to edge l_i are in different components (Γ_{ia} and Γ_{ib}) of the connectivity graph, then a splitting event of the reachable set occurs at the examined saddle point. Accordingly, two edges and a node are added to the splitting graph. If not, the steps 4D-E are skipped.

4D: for each face in the components Γ_{ia} and Γ_{ib}, the centroid solid angles of the areas over which $r(\mathbf{u})<r_i$, are determined (see Sec. 2D). The centroid solid angles are summed within each component to obtain $\Omega(\Gamma_{ia})$ and $\Omega(\Gamma_{ib})$.

4E: The transition probabilities associated with the splitting event are determined using (8).

5: the probabilities found in step 4F are propagated down the complete splitting graph to obtain the estimation.

A polyhedron with v vertices has O(v) edges and facets. Hence, the computational complexity of all steps is at most O(v). For almost all shapes, the number of stable poses and saddle points are only O(1), i.e. step 4 is repeated O(1) times. In sum, the complexity of the estimator is O(v).

References

[1] H. Hitakawa, “Advanced parts orientation system has wide application,” *Assembly Autom.*, Vol. 8, no. 3, pp. 147-150, 1988.

[2] B. Carlisle, K. Goldberg, A. Rao, and J. Wiegley, “A pivoting gripper for feeding industrial parts,” in *Proc. IEEE ICRA*, pp. 1650-1655, May 1994.

[3] H. Terasaki and T. Hasegawa, “Motion planning of intelligent manipulation by a parallel two-fingered gripper equipped with a simple rotating mechanism,” *IEEE Trans. Robot. Autom*., Vol. 14, no. 2, 207-219, Apr. 1998.

[4] A. Sudsang and L. E. Kavraki, “A geometric approach to designing a programmable force field with a unique stable equilibrium for parts in the plane,” in *Proc. IEEE ICRA*, Vol. 2, pp. 1079-1085, May 2001.

[5] A. Sudsang and L. Kavraki, “Part orientation with a force field: Orienting multiple shapes using a single field,” in *Proc. IEEE IROS* Vol. 1, pp. 208-213, Oct. 2001.

[6] K.-F Böhringer, B. R. Donald, L. Kavraki, and F . Lamiraux. “Part orientation with one or two stable equilibria using programmable vector fields,” *IEEE Trans. Robot. Autom.*, Vol. 16, pp. 157-170, Apr. 2000.

[7] T. H. Vose, P. Umbanhowar, and K. M. Lynch, “Sliding manipulation of rigid bodies on a controlled 6-DoF plate,” *Int. J. Robot. Res.*, Vol. 31, no. 7.,pp. 819-838, June 2012.

[8] M. Erdmann, M. T. Mason, and G. Vaněček, Jr., “Mechanical parts orienting: The case of a polyhedron on a table,” *Algorithmica*, Vol. 10 no. 2-4, pp. 226-247, Oct. 1993.

[9] K. M. Lynch, M. Northrop, and P. Pan, “Stable limit sets in a dynamic parts feeder,” *IEEE Trans. Robot. Autom.*, Vol. 18, no. 4, pp. 608-615, Aug. 2002.

[10] S. Akella and M. T. Mason, “Using partial sensor information to orient parts,” *Int. J. Robot. Res.*, vol. 18, no. 10, pp. 963-997, Oct. 1999.

[11] R. P. Berretty, M. H. Overmars, and A. F. van der Stappen, “Orienting polyhedral parts by pushing,” *Comput. Geom.*, Vol. 21, no. 1-2, pp. 21-38, Jan. 2002.

[12] R. P. Berretty, M. H. Overmars, and A. F. van der Stappen, “On fence design and the complexity of push plans for orienting parts,” in *Proc. 13th Ann. ACM Symp. Comput. Geom.*, pp. 21-29, Aug. 1997.

[13] J. Wiegley, K. Goldberg, M. Peshkin, and M. Brokowski, “A complete algorithm for designing passive fences to orient parts,” *Assembly Autom.*, Vol. 17, no. 2, pp. 129-136, 1997.

[14] R. P. Berretty, K. Goldberg, M. H. Overmars, and A. F. Van der Stappen, “Trap design for vibratory bowl feeders,” *Int. J. Robot. Res.*, Vol. 20, no. 11, pp. 891-908, Nov. 2001.

[15] O. C. Goemans and A. F. van der Stappen, “On the design of traps for feeding 3D parts on vibratory tracks,” *Robotica* Vol. *26*, no. 4, pp. 537-550, June 2008.

[16] M. Tao Zhang, K. Goldberg, G. Smith, R. P. Beretty, and M. Overmars, “Pin design for part feeding, “ *Robotica*, Vol. 19, no. 6, pp. 695-702, Sept. 2001.

[17] O. C. Goemans, K. Goldberg, and A.F. van der Stappen. “Blades: a new class of geometric structures for feeding 3d parts on vibratory tracks,” Proc. *IEEE ICRA*, pp. 1730–1736, May 2006.

[18] N. B. Zumel and M. A. Erdmann, “Nonprehensile manipulation for orienting parts in the plane,” *Proc. IEEE ICRA*, Vol. 3, pp. 2433-2439, Apr. 1997.

[19] D. R. Berkowitz and J. Canny, “Designing parts feeders using dynamic simulation,” in *Proc. IEEE ICRA*, Vol. 2, pp. 1127-1132, Apr. 1996.

[20] B. J. Slaboch, and P. A. Voglewede, “Underactuated Part Alignment System for Industrial Assembly Applications,” *J. Mechanisms Robot.* Vol. 5, 011006, 2012.

[21] M. Moll and M. A. Erdmann, “Manipulation of pose distributions,” *Int. J. Robot. Res.*, Vol. 21, no. 3, pp. 277-292, March 2002.

[22] M. Suresh, K. A. Jagadeesh, and P. A. Varthanan, “Determining the natural resting orientation of a part using drop test and theoretical methods,” *J. Manuf. Syst.*, Vol. 32, no. 1, pp. 220–227, Jan. 2013.

[23] P. Song, J. C. Trinkle, V. Kumar, and J. S. Pang, “Design of part feeding and assembly processes with dynamics,” in *Proc. IEEE ICRA*, Vol. 1, pp. 39-44, May 2004.

[24] G. Boothroyd and C. Ho , “Natural resting aspects of parts for automatic handling,” *J. Eng. for Industry*, vol. 99, pp. 314-317, May 1977.

[25] K. Goldberg, B. V. Mirtich, Y. Zhuang, J. Craig, B. R. Carlisle, and J. Canny “Part Pose Statistics: Estimators and Experiments,” *IEEE Trans. Robot. Autom.*, Vol. 15, pp. 849-857, Oct. 1999.

[26] J. Wiegley, A. Rao, and K. Goldberg, “Computing a statistical distribution of stable poses for a polyhedron,” in *Proc. 30th Annu. Allerton Conf. Commun., Contr., Comput., Univ. Illinois, Urbana-Champaign*, Oct. 1992.

[27] G. Boothroyd, A. H. Redford, C. R. Poli, and L. E. Murch, “Statistical distribution of natural resting aspects of parts for automatic handling,” *Manuf. Eng. Trans.*, Vol. 1, pp. 9 , 1972

[28] B. K. A. Ngoi, L. E. N. Lim, and S. S. G. Lee, “Analyzing the Probabilities of Natural Resting for a Component With a Virtual Resting Face,” *J. Manuf. Sci. Eng.*, vol. 120, pp. 468-471, May 1998.

[29] B. K. A. Ngoi and L. E. N. Lim, “Analysing the Natural Resting Aspect of a Component for Automated Assembly using the Energy Envelope Method,” *Int. J. Adv. Manuf. Technol.*, Vol. 12, no. 2, pp. 132-136, 1996.

[30] P. S. K. Chua and M. L. Tay, “Modelling the natural resting aspect of small regular shaped parts,” *J. Manuf. Sci. Eng*., Vol. 120, no. 3, pp. 540-546, Aug. 1998.

[31] S. S. G. Lee, B. K. A. Ngoi, L. E. N. Lim, and S. W. Lye, ”Determining the probabilities of the natural resting aspects of parts from their geometries,” *Assembly Autom.*, Vol. 17, no. 2, pp. 137–42, 1997.

[32] B. K. A. Ngoi, L. E. N. Lim, and J. T. Ee. „Analysis of natural resting aspects of parts in a vibratory bowl feeder—Validation of “drop test”" *Int. J. Adv. Manuf. Technol.* Vol. 13, no. 4, pp. 300-310, 1997.

[33] S. Udhayakumar, P.V. Mohanram, P. Keerthi Anand, R. Srinivasan, "Determining the most probable natural resting orientation of sector shaped parts" *Assembly Autom*., Vol. 33, no. 1, pp. 29 – 37, 2013.

[34] D. J. Kriegman, “Let them fall where they may: Capture regions of curved objects and polyhedral”. *Int. J. Robot. Res.* Vol. 16, no. 4, pp. 448-472, August 1997.

[35] G. Domokos, A. Á Sipos, and T. Szabó, T, “The mechanics of rocking stones: equilibria on separated scales,” *Math. Geosci.* Vol. 44 no. 1, pp. 71-89, Jan. 2012.

[36] A. Van Oosterom, A., and J. Strackee, “The solid angle of a plane triangle”. *IEEE Trans. Biomed. Eng.*, Vol. BME-30, no. 2, pp. 125-126, Feb. 1983.

[37] O. Mazonka, "Solid Angle of Conical Surfaces, Polyhedral Cones, and Intersecting Spherical Caps". *Cornell University Library Archive*, arXiv: 1205.1396, 2012.

[38] J. Arvo, "Fast random rotation matrices", in David Kirk, *Graphics Gems III*, San Diego: Academic Press Professional, pp. 117–120, ISBN 978-0-12-409671-4, 1992.

[39] A. Rodriguez, and A. Bowling, "Solution to indeterminate multipoint impact with frictional contact using constraints," *Multibody Syst. Dyn.* Vol. 28, pp. 313-330, 2012.

[40] J. Zhang, K. H. Johansson, J. Lygeros, and S. Sastry, “Zeno hybrid systems,” *Int. J. Robust Nonlin. Control*, Vol. 11 no. 5, pp. 435-451, 2001.

[41] A. D. Ames, H. Zheng, R. D. Gregg, and S. Sastry, “Is there life after Zeno? Taking executions past the breaking (Zeno) point,” in *Proc. Amer. Control Conf.*, pp. 6, June 2006.

[42] P.L. Várkonyi, "The secret of gambling with irregular dice: estimating the face statistics of polyhedra", extended abstract at the European Workshop on Comp. Geom. (EuroCG), Ein-Gedi, Israel, March 3-5, 2014.

[43] Goldstein H., *Classical Mechanics*, 2nd ed., Addison-Wesley. ISBN 0-201-02918-9, 1980.

Péter L. Várkonyi received the Ph.D. degree in architectural engineering from the Budapest University of Technology (Hungary) in 2006. Currently, he is associate professor at the Department of Mechanics, Materials and Structures of the same institution. His research focuses on rigid body dynamics in the presence of impacts or friction and its applications in various fields including robotics, biology and geomorphology.

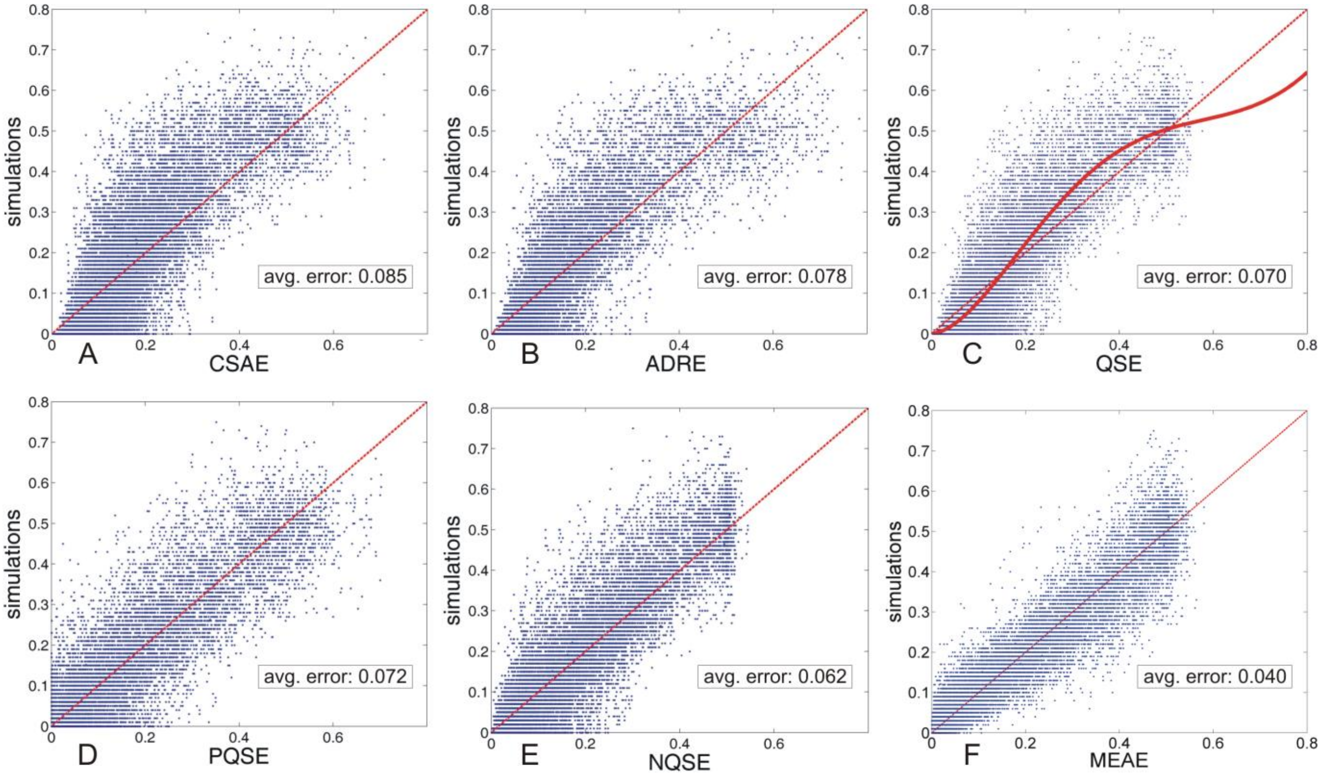

Fig. 4: comparison of pose statistics estimators. Each plot corresponds to an estimator discussed in the paper. Each individual dot in the plots shows the probability of coming to rest in a specific stable pose of a random polyhedron based on 100 simulations (vertical coordinate) vs. the prediction of the same probability provided by the estimator (horizontal coordinate). Points close to the main diagonal are indicators of a better estimator. Nevertheless, deviations from the main diagonal are caused partly by other factors (Fig. 2, Sec. 4.A). The solid curve in panel C is a parametric function fitted to the data points, which is used in Sec. 4.C.

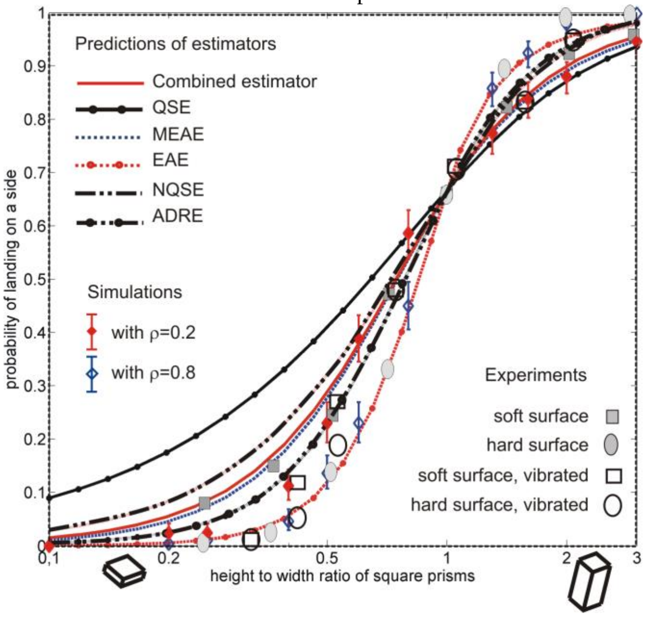

Fig. 5: Comparison of various predictions and measurements of the pose statistics of square prisms. The four sides of the prisms are treated as identical, similarly to the two bases. The diagrams show the probabilities of landing on the sides. Continuous curves show the predictions of various estimators. Diamonds with error bars are simulation results with the model of this paper and their 95% confidence intervals. Squares and circles indicate experimental results of [27] and [32]. Each point is based on 500 or more trials.