Algorithmic MEMS

Karl Friedrich Böhringer, *University of California at Berkeley, Berkeley, CA 94720*
Bruce Randall Donald, *Dartmouth College, Hanover, NH 03755*

Abstract

As improvements in fabrication technology for MEMS (microelectromechanical systems) increase the availability and diversity of these micromachines, engineers are defining a growing number of tasks to which they can be put. The idea of carrying out tasks using large coordinated systems of MEMS units motivates the development of automated, algorithmic methods for designing and controlling these groups of devices. We report here on progress towards *algorithmic MEMS,* taking on the challenge of design, control, and programming of massively-parallel arrays of microactuators.

We report on these developments in this focused survey paper, based on the research results originally reported in our 1994 paper [26] and developed further in [22, 20, 27, 23, 18, 21, 19, 26, 12, 15, 16, 9]. We describe how arrays of MEMS devices can move and position tiny parts, such as integrated circuit chips, in flexible and predictable ways by oscillatory or ciliary action. The theory of programmable force fields can model this action, leading to algorithms for various types of micromanipulation that require no sensing of where the part is. Experiments support the theory. We also report on how the theory and results can be generalized to the macroscopic scale [10, 11].

1 Introduction

The recent, rapid innovations and improvements in microfabrication technology have resulted in an evergrowing complexity of available MEMS devices, mandating the use of automated, algorithmic methods

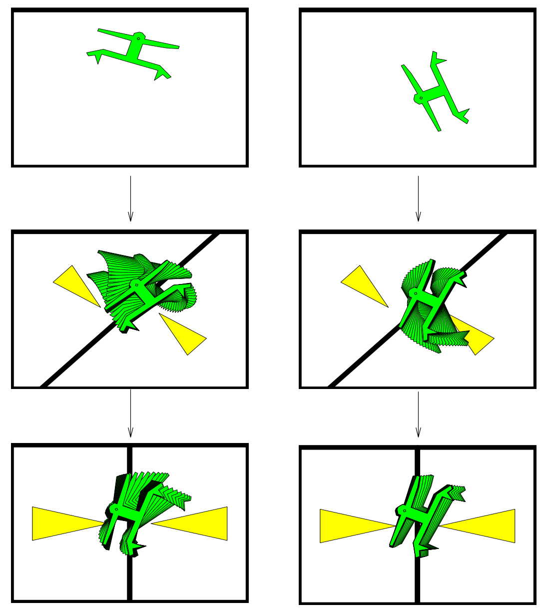

Figure 1: *Sensorless parts orienting using force vector fields: The part reaches unique orientation after two subsequent squeezes. There exist such orienting strategies for all polygonal parts. See URL* `http://www.cs.dartmouth.edu/~brd/demo/MicroManipulation` *for an animated simulation.*

for their design and control. This paper surveys our progress towards *algorithmic MEMS*, by taking on the challenge of design and control of massively-parallel micro actuator arrays. Our goal is to implement task-level, sensorless manipulation strategies with arrays of microfabricated actuators. The *theory of programmable vector fields*[1] [22] arguably represents the first systematic attack on massively-parallel, dis-

[1] The theory of programmable vector fields for micro manipulation tasks was introduced in [26].

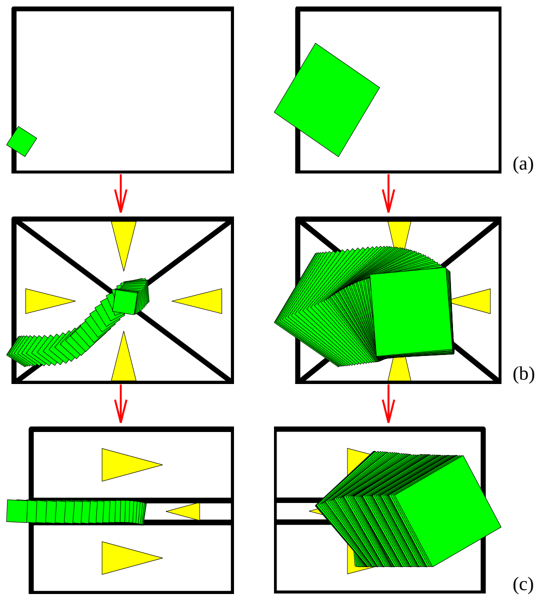

Figure 2: *Sensorless sorting using force vector fields: parts of different sizes are first centered and subsequently separated depending on their size.*

tributed manipulation based on geometric and physical reasoning. Applications such as parts-feeding can be formulated in terms of the force vector fields required. Hence, programmable force fields act as an *abstraction barrier* between applications requiring array micromanipulation and their implementation with MEMS devices. Such abstraction barriers permit hierarchical design, and allow application designs with greater independence from underlying device technology.

We are interested in the algorithmic content of MEMS control strategies. This paper surveys our work on showing how to quantify it, by analyzing the complexity of (1) computing a manipulation plan, (2) the generated plans (i.e., force field sequences) and (3) the individual fields. We survey our results on obtaining upper and lower complexity bounds on plans and fields, and in finding trade-offs between these different kinds of complexity. We review how to solve the problems of planning and control of micro actuator arrays for a wide range of tasks, and present a tutorial on algorithms that automatically generate manipulation plans for translating, orienting, centering, and sorting small parts [18, 22].

When a part is placed on the array, the programmed vector field induces a force and moment upon it. Over time, the part may come to rest in a dynamic equilibrium state. By chaining together sequences of vector fields, the equilibrium states of a part in the field may be cascaded to obtain a desired final state—for example, this state may represent a unique orientation or pose of the part (see Figures 1 and 2a-b). The resulting strategies work from any initial configuration (pose) of the part, require no sensing, and enjoy efficient planning algorithms. A system with such a behavior exhibits the *feeding property* [2]:

> A system has the *feeding property* over a set of parts $\mathcal{P}$ and a set of initial configurations $\mathcal{I}$ if, given any part $P \in \mathcal{P}$, there is some output configuration $\mathbf{q}$ such that the system can move P to $\mathbf{q}$ from any location in $\mathcal{I}$.

In Section 5 we present several force vector fields that possess the feeding property. Some of these fields are more powerful part feeders, as they allow the choice of a specific output configuration $\mathbf{q}$. Our work on programmable vector fields is related to nonprehensile manipulation [34, 62, 36]: in both cases, parts are manipulated without form or force closure.

Even though our fields are typically not smooth, it is possible to define a *potential* for certain fields (as a unique path integral), and to show that when an array ($\mathbb{R}^2$) potential exists, it always lifts to a smooth configuration space ($\mathbb{R}^2 \times \S1$) potential for the manipulated part [22]. Vector fields with potential have been shown to be theoretically well-suited for manipulation strategies, by classifying a sub-family of potential fields in which every part has stable equilibria [22]. Hence, such fields have been proposed for manipulation tasks in which we desire to cascade equilibria in order to uniquely pose or orient a part (for details see [26, 22, 20]). The theory of programmable vector fields permits calculation of stable equilibria for macroscopic parts in potential fields[2] (see Figure 15). In our experiments, we employed vector fields with potential for

[2]While this question has been well-studied for a point

parts-orientation and -posing tasks, and the theory was used to predict the equilibrium poses of specific parts (Figure 14). The poses predicted by the equilibrium analysis were observed in our experiments (Figures 13-left).

Perhaps surprisingly, the theory also predicts the existence of pathological fields which do not induce well-behaved equilibria. In particular, there exist perfectly plausible vector fields which induce *no* stable equilibrium in very simple parts. Although these fields are very simple, they result in limit cycles and quite complex behavior. We implemented such fields on an organic micro cilia array (see Section 3.2) [60, 27, 23]. Vector fields *without* potential were employed to cast parts into limit cycles, e.g. "infinite" rotation using a skew-symmetric squeeze field. The predicted behavior (Figure 16) for such "unstable" vector fields was also observed (Figure 13-right). This shows that rather complex—but potentially useful—behavior can be generated using very simple fields.

We survey recent experiments in implementing this theory using microfabricated actuator arrays [22, 27, 23]. In these experiments, strategies were programmed in a fine-grained SIMD fashion by specifying planar force vector fields. These programmable fields were implemented by moving the individual actuators in a cyclic, gait-like fashion. Motion in non-principal (e.g. diagonal) directions was effected by a pairwise coupling of the actuators to implement *virtual actuators* and *virtual gaits*, (analogous to the virtual legs employed by Raibert's hopping and running robots [53]). The tasks of parts-translation, -rotation, -orientation, and -centering were demonstrated using small IC dice.

The theory of programmable vector fields and virtual gaits gives a method for controlling a very large number of distributed actuators in a principled, geometric, task-level fashion. Whereas many control theories for multiple independent actuators break down as the number of actuators becomes very large, our systems will only become more robust as the actuators become denser and more numerous.

mass in a field, the issue is more subtle when lifted to a body with finite area, due to the moment covector. See [22] for details.

2 Research Growth

When we began working in MEMS in 1991, it was not immediately clear what fundamental algorithmic or computational problems arose in this new area. Even after obtaining our first results on the theory of programmable vector fields in 1993 [24], the received view in the community was that the chief computational issues in MEMS arose in (1) design, and (2) simulation. Indeed, it is these problems that motivated our initial efforts (see, e.g., [5, 6, 8, 7]): in 1990, Ralph Merkle of Xerox PARC and Kris Pister of Berkeley urged us to apply our results on the design and simulation [33] of snap fasteners to MEMS [52].

At that time, work on force fields for manipulation had been limited to the artificial potential fields first developed by Khatib, Koditschek, and Brooks.[3] While potential fields have been widely used in robot control [43, 44, 57, 54], micro-actuator arrays present us with the ability to *explicitly* program the applied force *at every point* in a vector field. Whereas previous work had developed control strategies with *artificial* potential fields, our fields are non-artificial (i.e., *physical*). Artificial potential fields require a tight feedback loop, in which, at each clock tick, the robot senses its state and looks up a control (i.e., a vector) using a state-indexed navigation function (i.e., a vector field). In contrast, physical potential fields employ no sensing, and the motion of the manipulated object evolves open-loop (for example, like a particle in a gravity field). This alone makes our application of potential field theory to micro-devices a different, and algorithmically challenging enterprise. Such fields can be composed using addition, sequential composition, or "parallel" composition by superposition of controls.

After our first paper appeared in January 1994 [26], it became clear that there exist deep and challenging algorithmic problems in MEMS and programmable vector fields, at the intersection of combinatorial algorithms, geometry, dynamical systems, and distributed

[3]A notable exception are the three-dimensional force fields used by Joffe and his collaborators at JPL [40], where AC magnetic fields are used to orient and assemble ferromagnetic parts – in 3D!

systems. A flurry of papers has emerged on new algorithms, new analysis, and new arrayed devices for programmable vector fields. During the next few years, the authors continued working with Noel MacDonald at the Cornell Nanofabrication Facility to develop and test new arrays of MEMS microactuators for programmable vector fields [19, 21, 19, 22, 20, 25]. At the same time, we worked with Greg Kovacs' group at the National Nanofabrication Facility at Stanford, to develop a control system for MEMS organic ciliary arrays, and to perform manipulation experiments with these arrays to manipulate IC dice using array-induced force fields [23, 27]. In parallel, we worked with Ken Goldberg at Berkeley and Vivek Bhatt at Cornell to generalize the theory to macroscopic devices, by developing algorithms for transversely vibrating plates in order to implement programmable vector fields [10, 11]. Finally, Danny Halperin, working with the authors, developed new upper and lower bounds, output-sensitive algorithms, and a precise computational-geometric analysis of the area bisectors arising in squeeze-field algorithms [15, 16, 17].

Other groups have also been active in developing new devices, analysis, and algorithms. Ken Goldberg worked with John Canny's group at Berkeley, to continue research on using vibrating plates for manipulation, showing that longitudinal vibrations can generate a rich vocabulary of programmable vector fields [56]. In addition, they developed sophisticated dynamic models and dynamic simulators for both MEMS devices and macroscopic vibrating plates [55]. Lydia Kavraki explored the power of continuous vector fields, and demonstrated an elliptical potential field capable of posing any part into one of two equilibrium states [41]. Peter Will and his colleagues at ISI have explored a number of different MEMS array designs, as well as algorithms and analysis for programmable vector fields [46, 31, 29, 30]. Andy Berlin, David Biegelsen, P. Cheung, and Warren Jackson at Xerox PARC have developed a novel MEMS microactuator array based on controllable air jets, with integrated control and sensing circuitry [4, 28]. Working at CMU, Bill Messner and Jonathan Luntz developed a small room whose floor is tiled with controllable, programmable, macroscopic wheels that can be driven and steered to manipulate large objects such as boxes [47]. Such a system can implement programmable vector fields. Together with Howie Choset, they analyzed the resulting dynamical system to obtain interesting results on controllability and programmable vector field algorithms based on conservative vs. non-conservative fields [48]. Daniela Rus has developed three-dimensional reconfigurable tilings of actuators that could deliver 3D programmable vector fields for manipulation and locomotion [45]. Working with the Berkeley Sensors and Actuators Group (BSAC), Karl Böhringer and Ken Goldberg explored how MEMS devices employing electrostatic fringing fields can be used to implement programmable force fields for parts manipulation and self-assembly [13, 14].

In short, there has been an explosion of new and exotic arrayed devices for both MEMS manipulation and macroscopic manipulation. The theory of programmable vector fields has been applied and extended to a variety of devices and systems. It is somewhat remarkable that the same analysis tools, fields, and algorithms apply to such a wide range of systems.

3 Experimental Devices and Setup

Several groups have described efforts to apply MEMS (micro electro mechanical system) actuators to positioning, inspection, and assembly tasks with small parts [51, 3, 38, 26, 46, for example]. However, the fabrication, control, and programming of micro-devices that can interact and actively change their environment remains challenging. Problems arise from

1. the limited range of motion and force that can be generated with microactuators,
2. the lack of sufficient sensor information with regard to manipulation tasks,
3. design limitations and geometric tolerances due to the fabrication process, and
4. uncertain material properties and the lack of adequate models for mechanisms at very small scales.

In the following subsections we describe two different types of actuator arrays: single-crystal silicon (SCS)

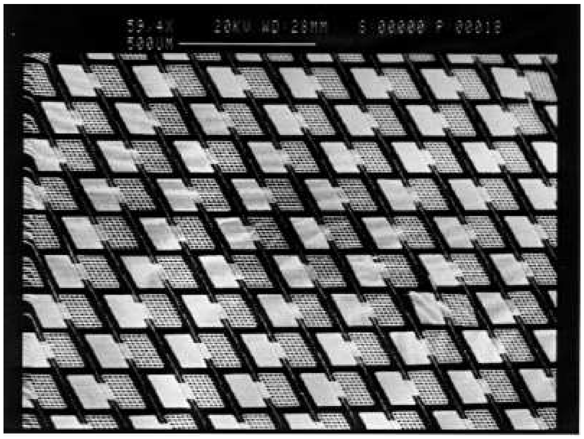

Figure 3: *Prototype* M-CHIP. *A large unidirectional actuator array (scanning electron microscopy). Each actuator is* $180 \times 240\,\mu m^2$ *in size. Detail from a 1* in^2 *array with more than 11,000 actuators. For more pictures on device design and fabrication see URL* `http://www.cs.dartmouth.edu/~brd/demo/MicroActuators`.

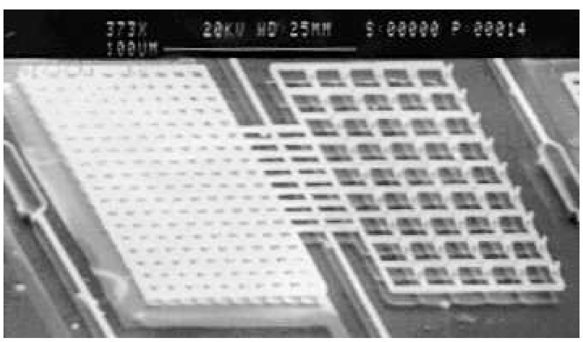

Figure 4: *Released asymmetric actuator for the* M-CHIP *(scanning electron microscopy). Left: Dense grid (*$10\,\mu m$ *spacing) with aluminum electrode underneath. Right: Grid with* $5\,\mu m$ *high poles.*

electrostatic actuators, and combined thermobimorph and electrostatic organic micro cilia.

3.1 Single-Crystal Silicon Electrostatic Actuators

We now survey recent work on single-crystal silicon electrostatic MEMS actuators, conducted in collaboration with Noel MacDonald at the Cornell Nanofabrication facility [19, 21, 19, 22, 20, 26, 25].

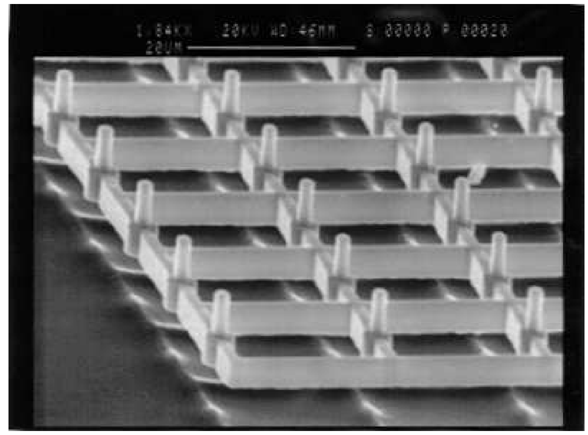

Figure 5: *Released* M-CHIP *actuator consisting of single-crystal silicon with* $5\,\mu m$ *high tips, suspended* $5\,\mu m$ *above the silicon substrate.*

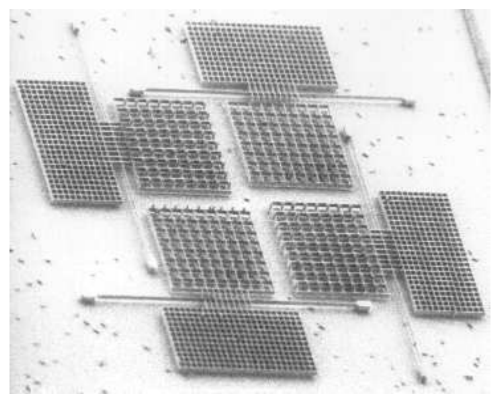

Figure 6: M-CHIP *prototype motion pixel consisting of actuators oriented in four different directions.*

The M-CHIP (Manipulation Chip, see Figure 3) is fabricated using a SCREAM (Single-Crystal Silicon Reactive Etching and Metallization) process developed in the Cornell Nanofabrication Facility [61, 59]. The SCREAM process is low-temperature, and does not interfere with traditional VLSI [58]. Hence it opens the door to building monolithic microelectromechanical systems with integrated microactuators and control circuitry on the same wafer.

The design is based on microfabricated torsional resonators [50, 49]. Each unit device consists of a rectangular grid etched out of single-crystal silicon suspended by two rods that act as torsional springs (Figure 4). The grid is about $200\,\mu m$ long and extends $120\,\mu m$ on each side of the rod. The rods are $150\,\mu m$ long. The current asymmetric design has $5\,\mu m$ high protruding tips on one side of the grid that make contact with an object lying on top of the actuator (Figure 5). The other side of the actuator consists of a denser grid above an aluminum electrode. If a voltage is applied between silicon substrate and electrode, the dense grid above the electrode is pulled downward by the resulting electrostatic force. Simultaneously the other side of the device (with the tips) is deflected out of the plane by several μm. Hence an object can be lifted and pushed sideways by the actuator.

Because of its low inertia (resonance in the high kHz range) the M-CHIP can be driven in a wide frequency range from DC to several $100\,kHz$ AC. The actuators need not be operated at resonance: They can also be servoed to periodically "hit" an object on top, hence applying both lateral and vertical forces. Calculations, simulations and experiments have shown that the force generated with a torsional actuator is approximately $10\,\mu N$, which corresponds to a force-per-area ratio of $100\,\mu N/mm^2$, large enough to levitate e.g. a piece of paper ($1\,\mu N/mm^2$) or a silicon wafer ($10\,\mu N/mm^2$).

Each actuator can generate motion in one specific direction if it is activated; otherwise it acts as a passive frictional contact. Figure 3 shows a small section of a unidirectional actuator array, which consists of more than 11,000 individual actuators. The combination and selective activation of several actuators with different motion bias allows us to generate various motions in discrete directions, spanning the plane (Figure 6). In initial manipulation experiments, small pieces of glass (size: several mm^2, mass: about $1\,mg$) were lifted within the motion range of the actuators (several μm) and pushed sideways by several $100\,\mu m$.

The fabrication process and mechanism analysis are described in more detail in [26, 25, 18].

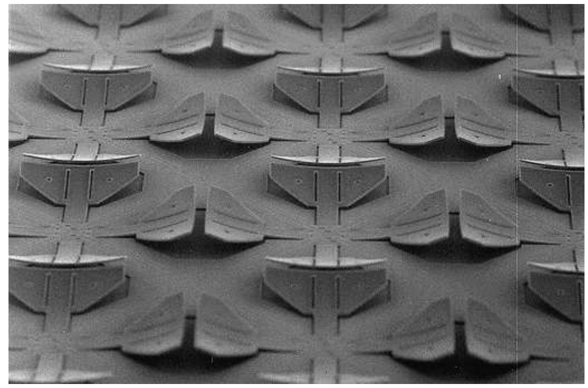

Figure 7: *Portion of a polyimide cilia array (SEM micrograph). Four orthogonally-oriented actuators are integrated into a motion pixel, which covers a surface area of approximately* $1.1 \times 1.1\,mm^2$.

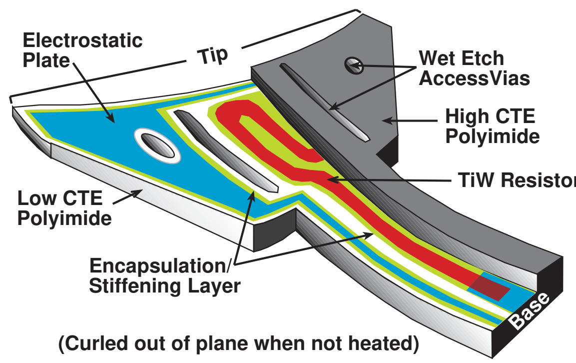

Figure 8: *Organic thermal and electrostatic microactuator. Half of the upper polyimide and silicon nitride encapsulation/stiffening layer are shown removed along the cilium's axis of symmetry to show details.*

3.2 Combined Thermobimorph and Electrostatic Polyimide Cilia Arrays

We now survey recent work on using MEMS organic cilliary arrays [60] for massively parallel microanipulation, conducted in collaboration with John Suh and Greg Kovacs at the Stanford University National Nanofabrication facility [23, 27].

At Stanford, surface micromachining techniques were used to create organic micro cilia arrays, consisting of polyimide as the primary structural material and aluminum as a sacrificial layer. The fabrication process (developed at the Center for Integrated Sys-

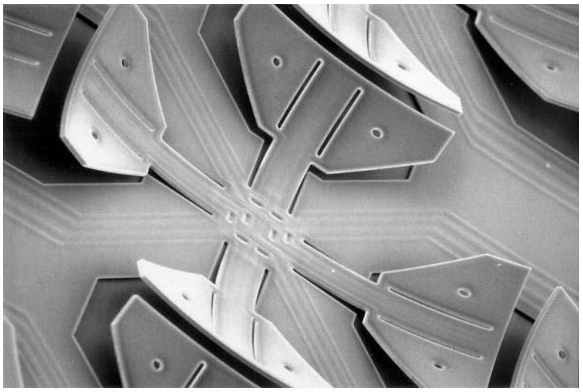

Figure 9: *Polyimide cilia motion pixel (SEM micrograph). Four actuators in a common center configuration make up a motion pixel. Each cilium is $430\,\mu m$ long and bends up to $120\,\mu m$ out of the plane.*

tems, Stanford University) was designed to be compatible with CMOS or BiCMOS circuits which could be pre-fabricated on a silicon substrate [60].

Actuator Cilia. Each cilium consists of two layers of polyimide with different thermal expansion coefficients. The cilium also contains a Titanium-Tungsten (Ti:W) resistive heater loop for thermal actuation, Aluminum electrodes for electrostatic (low-power) hold-down, and a silicon nitride encapsulation/stiffening layer (Figure 8). For a detailed description of the fabrication process see [60]. Vertical and horizontal displacements of the cilia tips are a function of the thermal mismatch in the actuator layers. For room temperature, these values can be calculated as $\delta_v \approx 120\,\mu m$ and $\delta_h \approx 20\,\mu m$ [60]. Inspection under the scanning electron microscope (SEM) has verified these calculations.

The lifting capacity of an actuator can be estimated as the force required to deflect the actuator's tip to the substrate. The actuator load capacity has been calculated as $F_l = 76\,\mu m$, which gives a force-per-area ratio of $4 \times 76\,\mu N/(1.1\,mm)^2 \approx 250\,\mu N/mm^2$.

Chip Layout. The cilia array is composed of cells (*motion pixels*, each $1.1 \times 1.1\,mm^2$) which contain four orthogonally-oriented actuators (Figure 9). On the current cilia chip, the motion pixels are arranged in an 8×8 array which occupies approximately $0.77\,cm^2$ of a $1\,cm^2$ die. The four actuators of each pixel are independently activated by four thermal and four electrostatic control lines. Four 8×8 chips are diced and packaged together to make a quad-shaped 16×16 cilia array device, with a total of 1024 cilia. The device itself is attached to a hybrid package which is placed on a heat sink and thermo-electric cooler (Peltier effect module). The total input power to the chip can exceed $4\,W$, and without active cooling the package can become very hot. To observe the experiments, a long working distance microscope is connected to a CCD camera, and a video cassette recorder is used to monitor and record both the movements of an individual cilium and the objects conveyed by the array.

Controller. The manipulation results described below were accomplished with the cilia array device interfaced to an IBM 486 personal computer. The PC provides speed control via the drive pulse frequency and directional control interactively via keyboard or mouse, or by actuator programs that can be specified using the MEMSA (MEMS Array) language[4] which we developed at Stanford. The control software including the MEMSA interpreter was written in PASCAL. Thermal and electrostatic control line signals are sent via the PC parallel port to D-type flip-flops which activate power transistors. Currently up to 4 cilia arrays can be controlled simultaneously by using a multiplexer with 2 address bits.

4 Low-level Control: Actuator Gaits

We now survey some basic concepts of actuator gaits using MEMS arrays for massively parallel microanipulation, developed in collaboration with Jonathan Suh and Greg Kovacs at the Stanford University National Nanofabrication facility [23, 27].

To induce motion on a part that is placed on the array, the cilia are actuated in a cyclic, gait-like fashion. In each cycle, the part is moved in a certain direction by the motion of the actuators that are in contact with

[4] MEMSA (named after MENSA) is the language for smart manipulation surfaces.

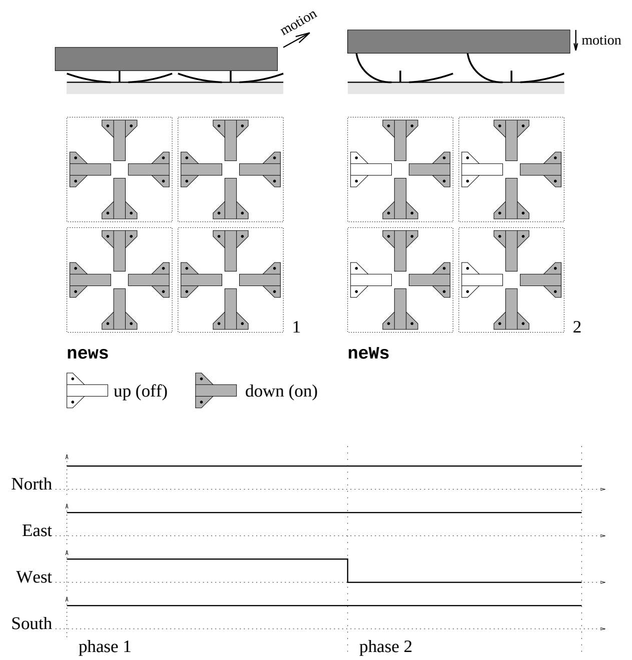

Figure 10: *Two-phase gait. The* `W` *actuator is repeatedly switched on and off, while the other actuators always remain on, resulting in a* `news neWs` *sequence.*

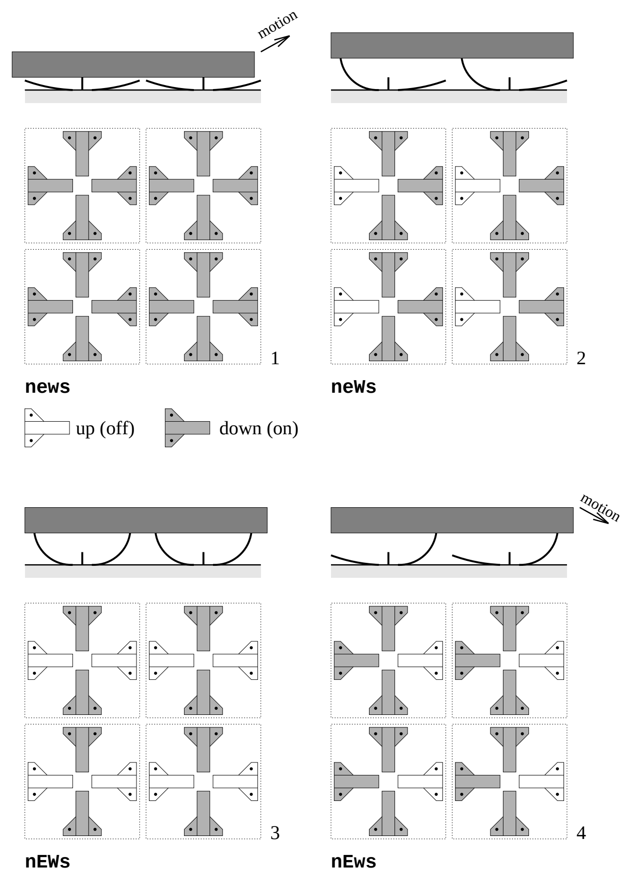

Figure 11: *Four-phase gait consisting of the four-pattern sequence* `news neWs nEWs nEws`.

it. The speed of the moving part depends on the (horizontal) displacement of the actuators per cycle as well as the frequency of cycle repetition. It also depends on the surface properties and weight of the moving part.

Task: Translation of Parts in Principal Directions. The simplest gait is the *two-phase gait*, in which all actuators of the same orientation repeatedly stroke the part while the remaining actuators are held down. Assuming that the orthogonal cilia within a motion pixel are oriented at the principal compass points, let us use capital letters `NEWS` to denote the North, East, West, and South actuators in the up position, and lower-case letters `news` to denote the actuators in the lowered position. Then the two-phase gait to effect motion in the East direction would be `news neWs` (see Figure 10).

The *four-phase gait* consists of four different actuation phases `news neWs nEWs nEws` such that motion is induced during upward as well as downward strokes of the cilia (Figure 11, see also [3]). Note that the forces exerted on the moving part depend on the state of the motion pixel: e.g. in the transition from phase 1 to phase 2 the cilium `W` moves up while the opposing cilium `e` remains down. We denote the lateral force exerted on the part in this configuration $f_{\mathtt{W},\mathtt{e}}$. Analogously, during transitions 2–3, 3–4, and 4–1 we observe lateral forces $f_{\mathtt{E},\mathtt{W}}$, $f_{\mathtt{w},\mathtt{E}}$, and $f_{\mathtt{e},\mathtt{w}}$, respectively. $f_{\mathtt{W},\mathtt{e}}$ and $f_{\mathtt{e},\mathtt{w}}$ are in positive x-direction, while $f_{\mathtt{E},\mathtt{W}}$ and $f_{\mathtt{w},\mathtt{E}}$ are negative. Furthermore, from this analysis it follows that $|f_{\mathtt{W},\mathtt{e}}| \gg |f_{\mathtt{e},\mathtt{w}}| \gg |f_{\mathtt{E},\mathtt{W}}| \approx |f_{\mathtt{w},\mathtt{E}}| \approx 0$. Hence we expect a relatively large motion step $\Delta x_{\mathtt{W},\mathtt{e}}$ during transition 1–2, and a smaller step $\Delta x_{\mathtt{e},\mathtt{w}}$ during transition 4–1, while during the other transitions the part remains at its current location. This behavior has been observed in our experiments, where $\Delta x_{\mathtt{W},\mathtt{e}}$ was measured between $3\,\mu m$ and $10\,\mu m$ depending on input power,

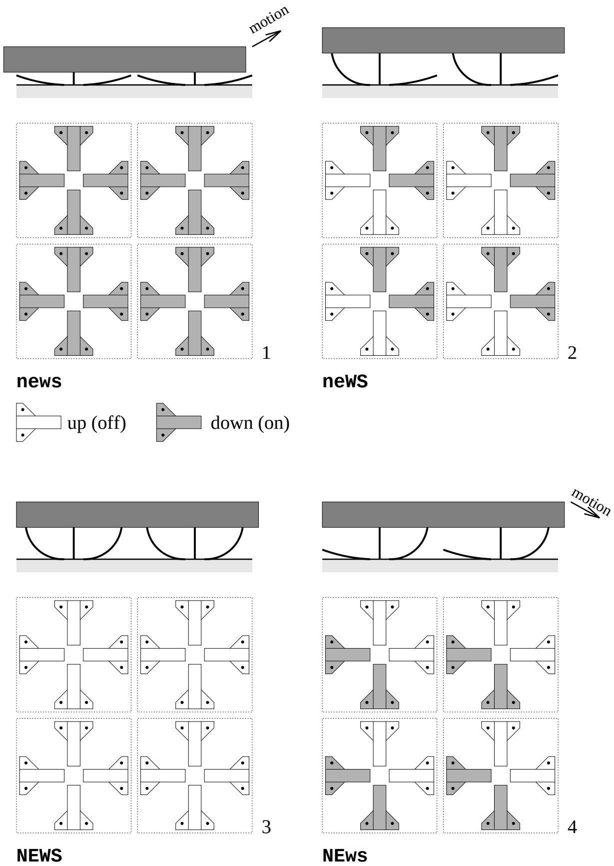

Figure 12: *Diagonal (virtual) gait consisting of the four-pattern sequence* `news neWS NEWS NEws`*. The* `N` *and* `E` *cilia, and the* `W` *and* `S` *cilia, are coupled to form virtual cilia in North-East and South-West directions.*

frequency, surface properties and weight of the part. $\Delta x_{\mathtt{W,e}}$ was usually about twice as large as $\Delta x_{\mathtt{e,w}}$.

Task: Translation of Parts in Arbitrary Directions. Motion in non-principal (e.g. diagonal) directions is effected by a pairwise coupling of two cilia of each pixel, implementing *virtual cilia* analogous to Raibert's concept of virtual legs for hopping and running robots [53]. Hence, several cilia can be coordinated to emulate a virtual cilium, which generates a force corresponding to the vector sum of its components. The diagonal gait to effect motion in the North-East direction would be `news neWS NEWS NEws` where the virtual cilia are `NE` and `WS`. Consequently, we obtain a *virtual gait* that moves the part in a diagonal direction. Note that in a section view through the array looking in the North-West direction, this gait looks virtually identical to the four-phase gait depicted in Figure 11.

Motion in arbitrary directions can be induced by alternating gaits that interleave principal (or virtual) gaits of different directions. For example, a translation at 25° from the x-axis requires motion in the y-direction and x-direction at a ratio of $\tan 25° \approx 1 : 2$. Our control software determines the exact alternation analogously to the Bresenham line scan algorithm [37], which rasterizes lines at arbitrary angles, resulting in different fields that are interlaced in time.

Experiments and Results. In collaboration with Jonathan Suh and Greg Kovacs at the Stanford University National Nanofabrication facility [23, 27], a large number of translation experiments have been performed in which two-phase and four-phase gaits were used to implement principal and virtual gaits. These experiments show that a first-order dynamical system models the device-part interaction well.[5] Therefore, when describing and predicting the motion of parts in force vector fields, we have based our theory on a first-order system (see Section 5).

Silicon chips were moved with a motion resolution of a few μm and speeds up to $200\,\mu m/sec$. Four-phase gaits proved more effective than two-phase gaits, because during the downward motion in the two-phase gait, the part tends to slip backwards. The four-phase gait avoids this effect, because other cilia hold the part in place during the transition 3–4. In the subsequent

[5] While experiments with our MEMS arrays indicate they may be modelled as first order (heavily-damped or essentially quasi-static) systems, it is easy to imagine future MEMS arrays that are highly dynamic and which would require a detailed analysis of dynamic equilibria. Indeed, our vibrating plates exhibit more dynamics, although empirically the quasi-static analysis has been adequate so far [10, 11]. Some experimental evidence for quasi-staticity is discussed in [23, 27], and the effect of dynamics on the quasi-static model is discussed in [20]. Research on dynamic analysis of programmable vector fields is an active research area; see, e.g., [55, 56, 31, 29, 30, 48].

downward motion in the transition 4–1, the part is also moved forward (Figure 11).

The diagonal gait also has the lowest power consumption (not considering electrostatic hold-down), due to the fact that its duty cycle for cilia hold-down is lowest (50 %), compared to 75 % for the principal four-phase gait, and 87.5 % for the two-phase gait.

As expected, diagonal (virtual) gaits induced the largest and fastest motion because all four cilia of each pixel were activated, whereas in principal gaits only two cilia are actively used, while the others have to be held down continuously (Figure 11).

5 High-level Control: Vector Fields

We believe that vector fields can be used as an *abstraction barrier* between applications requiring array micro-manipulation and MEMS devices implementing the requisite mechanical forces. That is, applications such as parts-feeding can be formulated in terms of the vector fields required. This then serves as a specification which the underlying MEMS device technology must deliver. Conversely, the capabilities of MEMS array technologies for actuation can be formulated in terms of the vector fields they can implement. For example, limitations in force magnitude are naturally expressed in vector field terms, as "small" vector fields. Restrictions in directional selectivity and magnitude control can also be manifested as restrictions on the vector field abstraction (resulting in discretization in orientation or modulus). This means that MEMS designers can potentially ignore certain details of the application process, and instead focus on matching the required vector field specification. Then, once the capabilities of MEMS actuator arrays were published as vector fields and tolerances, an application designer could look in a catalog to choose a device technology based on the field specification it promises to implement. This would free application engineers from needing to know much about process engineering, in the same way that software and algorithm designers often abstract away from details of the hardware. Such an abstraction barrier could permit hierarchical design, and allow application designs with greater independence from the un-

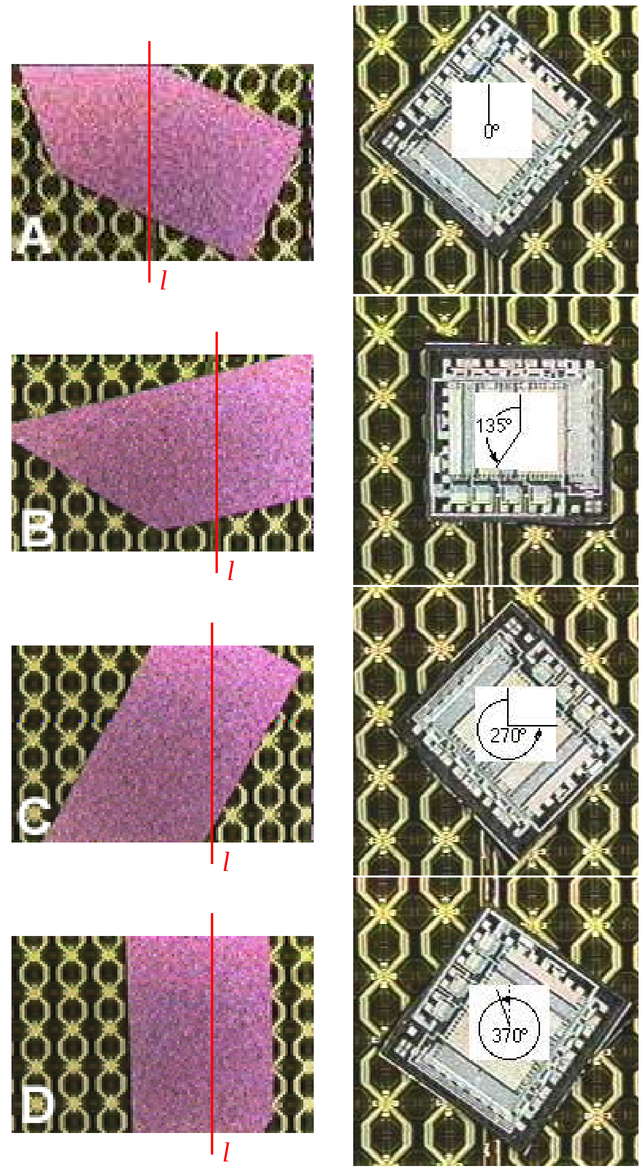

Figure 13: *Manipulation of silicon chips in programmable vector fields induced by a micro cilia array (microscope video images).*
Left: The chip is aligned with the vertical squeeze line l (marked by a dark line for clarification).
Right: Rotating a square-shaped chip counterclockwise in a skewed squeeze field.

derlying device technology. At the same time, abstraction barriers could allow MEMS array technologies to

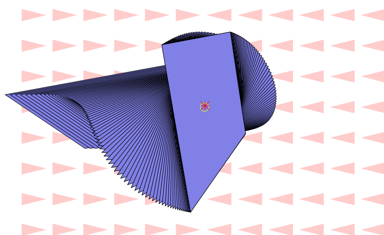

Figure 14: *Simulation of the alignment task with a squeeze field as shown in Figure 13-left.*

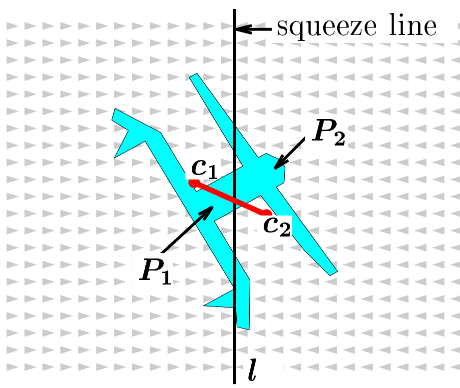

Figure 15: *Equilibrium condition: To balance force and moment acting on P in a unit squeeze field, the two areas P_1 and P_2 must be equal (i.e., l must bisect the part), and the line connecting the centers of area c_1 and c_2 must be perpendicular to the squeeze line l.*

be designed simultaneously with the (abstract) vector field control. This development pattern could be similar to the concurrent design of VLSI processors with their compilers, as is common in computer architecture.

We now survey some basic concepts and theory in high-level (algorithmic) control, and describe experiments to test the validity of the theory. The experimental results surveyed below in Sec. 5 were conducted in collaboration with Jonathan Suh and Greg Kovacs at the Stanford University National Nanofabrication facility [23, 27]. In this survey, we emphasize the geometric and algorithmic aspects of the problem, as well connections and synergy with experimental work on design, fabrication, and programming of MEMS arrays.

5.1 Squeeze Fields

In [26], the authors proposed a family of control strategies called *squeeze fields* and a planning algorithm for parts-orientation (see Figures 1 and 15).

Definition 1 [22] Given a straight line l, a *squeeze field* f is a two-dimensional force vector field in which, at each point, a unit force points perpendicularly towards l (on l the force is zero).

We refer to the line l as the *squeeze line*, because l lies in the center of the squeeze field.

Assuming quasi-static motion, an object will translate and rotate to an equilibrium configuration, as characterized in Figure 15. To predict the equilibria, we assume a uniform force distribution over the surface of P, which is a reasonable assumption for a flat part that is in contact with a large number of elastic actuators.

Definition 2 A part P is in *translational equilibrium* if the forces acting on P are balanced. P is in *orientational equilibrium* if the moments acting on P are balanced. *Total equilibrium* is simultaneous translational and orientational equilibrium.

Claim 3 [22] *Every squeeze field f (see Definition 5.1) has potential, of the form $U(\mathbf{z}) = \int_\alpha f \cdot ds$, where α is an arbitrary path to $\mathbf{z}$ from a fixed reference point. If $d_{\mathbf{z}}$ denotes the perpendicular distance of $\mathbf{z}$ from the squeeze line, then $U(d_{\mathbf{z}}) = |d_{\mathbf{z}}|$.*

Claim 4 [22] *Let P be a connected polygonal part with finite contact area and n vertices. Then in any squeeze field, P has $E = O(kn^2)$ orientation equilibria, where k is the maximum number of edges that a bisector of P can cross. If P is convex, then the number of equilibria is $O(n)$.*

Equilibria can be calculated numerically using the method in Figure 15: Given an arbitrary part at a fixed orientation, we translate it until its left and right sections have equal size. If the respective centers of gravity lie on a line perpendicular to the squeeze line, then the part is in equilibrium. For polygonal parts there exist analytical methods to compute the equilibria exactly (see [22] for a detailed algorithm and a derivation of the $O(kn^2)$ bound).

Claim 5 [22] *Let P be a polygon whose interior is connected. There exists a sensorless alignment strategy consisting of a sequence of squeeze fields that uniquely orients P up to symmetries.*

Proof: Claim 4 states that a squeeze field brings a polygon P into one of $E = O(kn^2)$ orientation equilibria. We define the *squeeze function* as this mapping from original orientation to equilibrium orientation. Hence, from Claim 4 it follows that the image of the squeeze function is a set of E discrete values. Given such a squeeze function, Goldberg has presented an algorithm for sensorless manipulation of polygons [39] that constructs an orienting strategy with $O(E)$ steps in $O(E^2)$ time. The output of this algorithm is a sequence of squeeze directions. When the corresponding squeeze fields are applied to the part P, the set of possible orientations is successively reduced until a unique orientation (up to symmetry) is reached. For details see [22] and [39]. □

Task: Orienting and Aligning Parts

If a part is placed in a squeeze field, it will translate and rotate until a stable equilibrium is reached (Claim 4). Parts may exhibit several equilibria, hence after one squeeze the part orientation may be ambiguous. This ambiguity can be removed with the strategies of Claim 5: by executing a sequence of squeezes at particular angles, the part is uniquely oriented (see Figure 1).

Experiments and Results. The long, thin part depicted in Figures 13-left and 14 exhibits a unique stable equilibrium (modulo 180° field symmetry). When placed in a squeeze field, its longitudinal axis aligns with the squeeze line. This dynamic process is predicted by simulation in Figure 14, and verified in experiment (see Figure 13-left). This part alignment experiment has also been performed with similar results for several other small pieces of glass and silicon of a few *mm* length and several *mg* of mass.

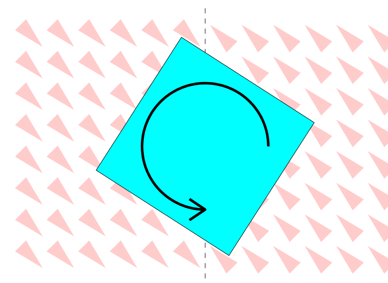

Figure 16: *Unstable square-shaped part in a skewed squeeze field ($\epsilon = -1$). The square with center on the squeeze line will rotate indefinitely. Moreover, it has* no *stable equilibrium in this field.*

5.2 Skewed Squeeze Fields

Definition 6 [22] A *skewed field* f_S is a vector field given by $f_S(x, y) = -\mathrm{sign}(x)(1, \epsilon)$, where $0 \neq \epsilon \in \mathbb{R}$.

Claim 7 [22] *No skewed squeeze field has a potential.*

In a skewed squeeze field, it is easy to find a circular path along which the work integral is non-zero (e.g., along a circle with center on the squeeze line).

Claim 8 [22] *A skewed field induces* no *stable equilibrium on a disk-shaped part (for all $\epsilon \neq 0$).*

Force equilibrium is only possible if the center of the disk coincides with the squeeze line. In this position the disk experiences a non-zero moment if $\epsilon \neq 0$.

Claim 9 *A diagonally skewed field ($\epsilon = \pm 1$) induces* no *stable equilibrium on a square-shaped part.*

For a proof see [22].

Task: Rotating Parts

According to Claims 8 and 9, certain parts will rotate indefinitely in skewed squeeze fields (Figure 16). Note that even though our cilia device has more degrees of freedom, two areas of constant force are sufficient to implement a skewed field, resulting in a very simple task-level rotation strategy. In particular, the rotation algorithm resulting from the application of skew-symmetric squeeze fields is considerably simpler than rotation algorithms proposed in the MEMS literature (for example, the vortices suggested by Fujita [38] or by Liu and Will [46]). Vortices require at least four areas of the array to be pushing in different directions. That is, vortices can be implemented using four triangular or rectangular regions, upon each of which the vector field is constant. Skewed fields perform the same task with only two regions of constant force.

Experiments and Results. Figure 13-right shows video frames of a $3 \times 3\, mm^2$ IC chip rotating on the squeeze line of a skewed field. During the experiments of approximately 10 min, several full rotations of the part were performed.

5.3 Radial Fields

Definition 10 A radial field f is a two-dimensional force vector field such that $f(\mathbf{z}) = -\mathbf{z}/|\mathbf{z}|$ if $\mathbf{z} \neq 0$, and $f(\mathbf{0}) = \mathbf{0}$.

Claim 11 **[22]** *A radial field has a potential, $U(\mathbf{z}) = |\mathbf{z}|$.*

Claim 12 **[22]** *Given a polygonal part P in a radial field f, there exists a unique* pivot point *v of P such that P is in equilibrium if and only if v coincides with the center of the radial field.*

Claim 13 **[22]** *Let P be a polygonal part with n vertices, and let k be the maximum number of edges that a bisector of P can cross. There are at most $E = O(kn)$ stable equilibria in a field of the form $R + \delta S$ if S is a squeeze field, and δ is sufficiently small and positive.*

Proofs of the previous claims, and a numerical algorithm to compute the pivot point is given in [22]. Note that Claim 13 results in strategies for unique part posing in $O(E^2) = O(k^2n^2)$ steps.

Task: Centering Parts

Radial fields can be used to center a part. With the current four-quadrant cilia device, we have implemented an approximation of an ideal radial field similar to the field in Figure 2-b. Note that this approximate radial field has a potential.

Experiments and Results. Small silicon and glass parts were centered using our cilia device. In this experiment, high positioning accuracy (in the μm range) was hard to achieve, because the center of the radial field coincides with the the location of the dice edges. Manual packaging of the four cilia chips leaves small gaps and non-planarities at these junctions. The next generation cilia device will overcome this problem, because it will allow us to implement the radial field with a single chip. Furthermore, because of its full pixel-wise programmability, the new chip will allow us to closely approximate ideal radial fields.

6 Polygon Bisectors and Force Equilibria

In the previous section we introduced vector fields for high-level control of micro actuator arrays. In particular, it was stated that in a squeeze field, polygonal parts have a (usually very small) finite number of orientation equilibria (see Claim 4). Because of this result, squeeze fields play a key role in manipulation strategies. In this section, we survey recent research on the combinatorial, geometric, and algorithmic properties of squeeze fields, by analyzing the *area bisectors* of a part. The results in Sec. 6 were obtained in collaboration with Danny Halperin [15, 16, 17].

There is a direct relationship between equilibria in squeeze fields and area bisectors. Recall Figure 15: a part is in force equilibrium if and only if the squeeze line bisects the part into two sections of equal area.

Definition 14 **[15]** Let P be a polygon in the plane, possibly with holes, and having n vertices in total. We denote by V the set of vertices of P. For a directed line λ in the plane, we denote by $h_l(\lambda)$ (resp. $h_r(\lambda)$) the open half-plane bounded by λ on the left- (resp. right-) hand-side of λ. The line λ is an *area bisector* of P if the area of $P \cap h_l(\lambda)$ is equal to the area of $P \cap h_r(\lambda)$.

A line λ partitions V into three sets (two of which may be empty): $V \cap h_l(\lambda)$, $V \cap \lambda$, and $V \cap h_r(\lambda)$. We say that two area bisectors of P are *combinatorially distinct* if the partitioning of V as above induced by the two bisectors is different. We say that two area bisectors of P are *combinatorially equivalent* if they induce the same partitioning of V. We assume that the polygon P is connected, and non-degenerate in the sense that the complement of P has the same boundary as P.

An obvious upper bound on the number of distinct area bisectors of a polygon with n vertices is $O(n^2)$—each combinatorial equivalence class of area bisectors is determined by a pair of vertices of the polygon. In Section 6.2 we describe how a polygon with n vertices can have $\Omega(n^2)$ distinct area bisectors [15]. (Note that the polygon in this construction is *simple*.)

We review an output-sensitive algorithm [15] for computing an explicit representation of all the area bisectors of a given polygon, by constructing the *bisector curve* β defined in a plane dual to the plane containing the polygon: the curve β is the union of points dual to area bisectors in the primal plane. The algorithm proceeds by constructing the *zone* of the curve β in an *arrangement of lines* [35] in the dual plane, where each line of the arrangement is the dual of a vertex of the polygon. A sketch of the algorithm is given in Section 6.3.

Area bisectors were considered by Díaz and O'Rourke [32]. However, their focus is on the continuous version of the *ham-sandwich cut* problem, and of a problem they introduce of *orthogonal four-sections*; see [32] for more details. The problem discussed here can be viewed as a continuous version of the well-known *k-set* problem [35].

In the remainder of this section, we analyze the number of area bisectors, and hence bound the number of force equilibria. The algorithm for computing area bisectors can be used as a preliminary step in designing alignment plans (see Section 6.4).

6.1 Properties of Area Bisectors

In this section we review several properties of area bisectors of polygons. Some of the proofs have been omitted here; they can be found in [17].

Lemma 15 **[15]** *Let P be a non-degenerate polygon with n vertices. (1) There exist $O(n^2)$ combinatorially distinct ways in which a line can partition P. (2) Let A and B be the intersections of an area bisector λ with the boundary of the convex hull of P. As the slope of λ varies from $-\infty$ to $+\infty$, A and B progress monotonically counterclockwise on the boundary of the convex hull of P. (3) For every slope $\bar{x}$ there exists a unique bisector λ of P with slope $\bar{x}$.*

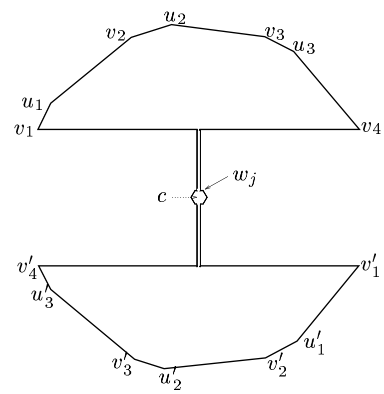

Figure 17: *A simple polygon with n vertices that has $\Omega(n^2)$ combinatorially distinct bisectors.*

Lemma 16 **[15]** *Let P be a polygon with n vertices. Let s be a point in $\mathbb{R}^2$ and let λ be a line that intersects r edges of P. The area bisectors of P that are combinatorially equivalent to λ and pass through s are determined by the roots of a polynomial equation of degree r.*

It can be shown [17] that the bisectors of a polygon P can be described by a piece-wise algebraic curve, where each piece is described by a polynomial whose degree depends on the number of edges of P intersected by the corresponding bisectors.

6.2 Lower Bound

As argued above, a polygon with n vertices can have at most $O(n^2)$ combinatorially distinct area bisectors. In this section we present an example of a simple polygon with n vertices where the bound $\Omega(n^2)$ is attained [15].

Consider Figure 17. All the vertices v_i, v'_i, u_i and u'_i lie on a circle whose center is at c. The vertices w_j lie very close to c on a small circle whose center is c as well, along two convex polygonal chains.

We fix an integer m (that we will determine later; for the polygon in the figure $m = 3$). The distance between the vertices v_i and v_{i+1} is the same for $i = 1, \ldots, m$, and it is the same as the distance between v'_i and v'_{i+1} for $i = 1, \ldots, m$. The area of all triangles $v_i u_i v_{i+1}$ for $i = 1, \ldots, m$ is the same and is equal to the area of all triangles $v'_i u'_i v'_{i+1}$ for $i = 1, \ldots, m$. There are $2m$ vertices w_j near c and they are equally spaced on a small circle centered at c. As can be easily verified, for every pair of vertices v_i and v'_i, there is a bisector passing through these points that passes also through the center point c. We next claim that as we rotate the bisector from v_i to v_{i+1} it will move off the center c and sweep m vertices w_j. The reason is that the angle $\angle u_i v_i v_{i+1}$ is larger than the angle $\angle u'_i v'_i v'_{i+1}$. Hence, as the bisector rotates, it will proceed 'faster' on the bottom part of our polygon than on the top part and therefore will sweep half of the vertices w_j on its way. Finally m is chosen such that (roughly) $n = 6m + 8$. The number of distinct area bisectors is evidently $\Omega(m^2) = \Omega(n^2)$.

6.3 Output-Sensitive Algorithm

It is convenient to study the algorithmic problem in a dual plane: a line $y = 2\bar{x}x - \bar{y}$ in the primal plane is transformed into the point $(\bar{x}, \bar{y})$ in the dual plane. A point (x, y) in the primal plane is transformed into the line $\bar{y} = 2x\bar{x} - y$ in the dual. The dual of an object o will be denoted by o^*. If O is a set of objects in the plane, O^* will denote the set of dual objects.

Let P be a polygon with n vertices as defined in the Introduction, namely connected, non-degenerate and possibly with holes. In the dual plane every vertex v of P is transformed into a line v^* which is the collection of all points dual to lines in the primal plane that pass through v.

For any given direction there is a unique area bisector. We denote the oriented bisector of P that makes an angle θ with the positive x-axis by $B(\theta)$, and (because of symmetry) confine ourselves to the range $[-\pi/2, \pi/2)$ for θ. We denote the collection of points dual to area bisectors of P in that range by β. Note that any θ (besides $-\pi/2$) corresponds to an $\bar{x}$-coordinate in the dual plane.

The curve β is a piece-wise algebraic and $\bar{x}$-monotone curve (this is proved in [17]). We call β the *bisector curve* of P, as it gives a complete specification of all the area bisectors of the polygon P. We denote by κ the number of maximal connected algebraic pieces in β, where the function describing each piece is defined by the fixed set of edges that the corresponding set of bisectors cross. In this section we describe an output-sensitive algorithm from [15] to compute β. Since we aim for output-sensitivity, we cannot afford to compute the entire arrangement $\mathcal{A}(V^*)$ whose complexity may be $\Omega(n^2)$. We will discover the maximal pieces of β in their order along β, using two primitive operations: *ray shooting* among the lines V^*, and intersection of an algebraic curve with a straight line.

We choose an arbitrary direction $\theta_0 \in [-\pi/2, \pi/2)$ and look for the area bisector of P in that direction. This requires $O(n \log n)$ time since the polygon may have holes. Next, we obtain the set of edges crossed by $B(\theta_0)$. We denote by $E(\theta)$ the set of edges crossed by $B(\theta)$. The set $E(\theta_0)$ determines a function $\rho := \rho(\theta_0)$ describing the bisector curve β in a neighborhood of θ_0, as long as the set of edges crossed by the bisector does not change. In the dual plane the function ρ describes the curve β as long as we do not leave the face of $\mathcal{A}(V^*)$ which contains the point $p := (B(\theta_0))^*$.

Our next step is to construct the face f that contains the point p in $\mathcal{A}(V^*)$. We describe this procedure assuming f is bounded; the extension to unbounded faces is straightforward. We prepare in advance a data structure $R(V^*)$ that supports efficient ray shooting among the lines V^*. We shoot a ray from p in the upward vertical direction and identify the line $\ell_1 \in V^*$ supporting the edge of f above p. See Figure 18(a). We next proceed in clockwise direction along the boundary of f. From the point $p_1 \in \ell_1$ we shoot a ray in $\mathcal{A}(V^*)$ along ℓ_1, and identify the line ℓ_2 supporting the next edge on the boundary of f, and so on until we have returned to ℓ_1 and thus have identified the entire face f.

Now we determine the maximal connected pieces of $f \cap \beta$. For each edge on boundary of f we compute the intersection of its supporting line v^* with β. This computation is equivalent to finding the bisectors that pass through the vertex v, and intersecting the edges

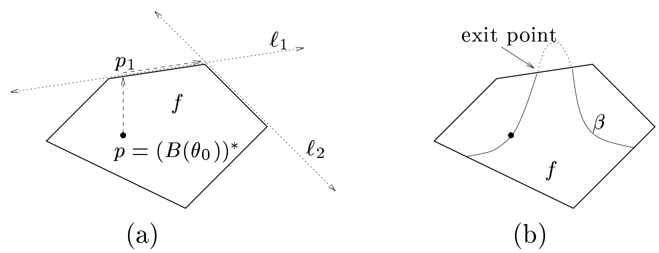

Figure 18: *Ray shooting to determine the face f containing p (a), and then finding the maximal pieces of β inside f and its exit points from f (b).*

in $E(\theta_0)$. By Lemma 16, this reduces to solving a polynomial equation of degree r, where r is the number of edges in $E(\theta_0)$. We denote the time required to find these roots by $\psi(r)$.

We order the resulting intersections along the $\bar{x}$-axis. Since the curve β is $\bar{x}$-monotone, this ordered list of intersections provides a description of the curve β inside f, and indicates what are the neighboring faces that β crosses. We mark each of these additional faces by the point where β crosses out of f. We call each such point an *exit point*. See Figure 18(b).

Since f has already been constructed, we know for each exit point of β the line that contains it. Therefore we can construct each new face using ray shooting queries and proceed as above. We keep a data structure that describes all the faces of $\mathcal{A}(V^*)$ that have already been constructed so that we do not construct the same face twice.

The algorithm stops when we have identified all the intersection points of β with lines in V^*, and so we have also identified the *zone* of β in $\mathcal{A}(V^*)$, namely all the face of $\mathcal{A}(V^*)$ crossed by β.

Further details on the algorithm can be found in [17]. We summarize the algorithmic result in the following theorem.

Theorem 17 [15] *Let P be a non-degenerate polygon (possibly with holes) with n vertices, and such that any line crosses at most r edges of P. For any $\varepsilon > 0$ we can find a complete specification of the area bisectors of P in time $O(\kappa^{2/3}n^{2/3+\varepsilon} + \kappa\alpha(\kappa)\psi(r))$, where κ and $\psi(r)$ are as defined above, and $\alpha()$ is the functional inverse of Ackermann's function. If P is rectilinear, then the algorithms runs in time $O(\kappa^{2/3}n^{2/3+\varepsilon})$. The space required by the algorithm is $O(\kappa^{2/3}n^{2/3+\varepsilon})$.*

6.4 Moment Equilibria and Alignment Plans

Recall from the beginning of this section that bisectors correspond to *force* equilibria of P in a squeeze field. For *total* equilibrium, the moment acting on the part has to be taken into account as well. In particular, not all of the force equilibrium configurations will be moment equilibria. For each maximal piece b of the bisector curve β there exists only a finite number of moment equilibria (we omit the proof here):

Claim 18 [15] *Let P be a polygon whose interior is connected. Let λ be a bisector of P that intersects r edges of P. There exist $O(r)$ lines λ' that are combinatorially equivalent to λ such that P is in total equilibrium when λ' coincides with the center line of a squeeze field.*

It follows that a squeeze field induces a finite number of total equilibria on a polygonal part P.

Much remains to be done. We have found that in practice, no non-symmetric part has more than 4 total equilibria (i.e., force and moment). Finding tight bounds on the number of total equilibria—that is, taking moment into account—will be critical in the future.

7 Conclusions

Table 1 summarizes fields and algorithms for manipulation tasks with programmable vector fields, and includes some additional recent results.

Less difficult tasks such as translation can be achieved with relatively simple fields and without any planning. More complex tasks such as centering or unique orienting require increasingly complex fields. However, planning complexity is e.g. higher for sequences of squeeze fields, and lower for the more complex combined radial + squeeze fields (for a thorough discussion of these fields see [22]). This illustrates a

task	field(s)	complexity		
		fields	planning	plan steps
translate	constant	constant magnitude and direction	—	1
center	radial	constant magnitude, continuous directions	—	1
uniquely orient	seq. of squeezes	piecewise constant mag. and dir.	$O(k^2n^4)$	$O(k\,n^2)$
	inertial	smooth magnitude piecewise constant dir.	$O(1)$	$O(1)$
uniquely pose	seq. of radial+squeeze	piecewise continuous mag. and dir.	$O(k^2n^2)$	$O(k\,n)$
	elliptic	smooth mag. and dir.	$O(1)$	$O(1)$
	UFO	continuous mag. and dir.	—	1

Table 1: *Fields and algorithms for manipulation tasks with programmable vector fields.*

tradeoff between mechanical complexity (the dexterity and controllability of actuator arrray elements) and computational complexity (the algorithmic difficulty of synthesizing a strategy).

Universal Feeder-Orienter (UFO) Devices. Our research leads to the question about the existence of a "universal feeder-orienter" (UFO) device that uniquely poses a part without the need of a clock, sensors, or programming [22, 20]. It was shown in [22] that every connected polygonal part P with n vertices has a finite number of stable orientation equilibria when P is placed into a squeeze field S. Based on this property we were able to generate manipulation strategies for unique part alignment. We then showed that by using a combined radial and squeeze field $R+\delta S$, the number of equilibria can be reduced to $O(k\,n)$. Using elliptic force fields $f(x,y) = (\alpha x, \beta y)$ such that $\alpha \neq \beta$ and $\alpha, \beta \neq 0$, this bound can be reduced to two [42, 41]. An "inertial" squeeze field $f(x,y) = (-\mathrm{sign}(x)x^2, 0)$ uniquely orients a part modulo field symmetry π [23]. In a stable equilibrium, the part's major principal axis of inertia lines up with the squeeze line to minimize the second moment of inertia.

Does there exist a *universal field* that, for every part P, has only one unique equilibrium (up to part symmetry)? Such a field could be used to build a *universal parts feeder* [1] that uniquely positions a part without the need of a clock, sensors, or programming.

In [22], we proposed a combined radial and "gravitational" field $R + \delta G$ which might have this property. δ is a small positive constant, and G is defined as $G(x,y) = (0,-1)$. This device design is inspired by the "universal gripper" in [1]. Such a field could be obtained from a MEMS array that implements a unit radial force field. Instead of rectangular actuators in a regular grid, triangular actuators could be laid out in a polar-coordinate grid. The array could then be tilted slightly to obtain the gravity component. Hence such a device would be relatively easy to build. Alternatively, a resonating speaker, or a vibrating disk-shaped plate that is fixed at the center, might be used to create a radial force field. Extensive simulations show that for every part we have tried, one unique total equilibrium is always obtained. We are working toward a rigorous proof of this experimental observation.

Algorithmic MEMS In this paper, we have surveyed a family of ideas in Algorithmic MEMS, relating to programmable force fields. Naturally, in such a young and dynamic field, we expect a myriad of new geometric and algorithmic problems to emerge in the next few years.

Acknowledgements

We are very grateful to our collaborators and friends Noel MacDonald, John Suh, Greg Kovacs, Ken Goldberg, and Danny Halperin for sharing with us the research surveyed in this paper.

The authors would also like to thank Steve Glander, Robert Darling, Christopher Storment, and all the members of the Stanford Transducers Lab; the members of Noel MacDonald's research team; staff and users of the Cornell Nanofabrication Facility, the members of the Cornell Robotics and Vision Lab (CSRVL), the members of Donald Lab at Dartmouth, Ken Goldberg and Lydia Kavraki for fruitful discussions, and Jean-Claude Latombe for his hospitality during our stay at the Stanford Robotics Laboratory.

Support for our research was provided in part by the NSF under grants no. IRI-8802390, IRI-9000532, IRI-9201699, IRI-9530785, and by a Presidential Young Investigator award to Bruce Donald, in part by NSF/ARPA SGER no. IRI-9403903, in part by an NSF Challenges in Computer and Information Science and Engineering (CISE) grant No. CDA-9726389, and in part by the AFOSR, the Mathematical Sciences Institute, Intel Corporation, and AT&T Bell laboratories.

References

[1] Tamara Lynn Abell and Mike Erdmann. A universal parts feeder, 1996. Personal communication / in preparation.

[2] Srinivas Akella, Wesley H. Huang, Kevin M. Lynch, and Matthew T. Mason. Planar manipulation on a conveyor by a one joint robot with and without sensing. In *International Symposium of Robotics Research (ISRR)*, 1995.

[3] Manabu Ataka, Akito Omodaka, and Hiroyuki Fujita. A biomimetic micro motion system. In *Transducers — Digest Int. Conf. on Solid-State Sensors and Actuators*, pages 38–41, Pacifico, Yokohama, Japan, June 1993.

[4] D. Biegelsen, W. Jackson, A. Berlin, and P. Cheung. Air jet arrays for precision positional control of flexible media. In *Int. Conf. on Micromechatronics for Information And Precision Equipment (MIPE'97)*, Tokyo, Japan, July 1997.

[5] Karl-Friedrich Böhringer. Computational aspects of the design of micro-mechanical hinged structures. In *AAAI Fall Symposium Series — Design from Physical Principles, Working Notes*, Boston, MA, October 1992. Abstract. .

[6] Karl-Friedrich Böhringer. Application of computational topology to the design of microelectromechanical structures. In *Proceedings of the NSF Design and Manufacturing Systems Grantees Conference*, Charlotte, NC, January 1993. National Science Foundation. .

[7] Karl-Friedrich Böhringer. A computational approach to the design of micromechanical hinged structures. In *Proceedings of the ACM/SIGGRAPH Symposium on Solid Modeling and Applications*, Montréal, Québec, Canada, May 1993. ACM Press. Extended abstract. .

[8] Karl-Friedrich Böhringer. Computational aspects of the design of microelectromechanical structures. In *ICRA Workshop on Geometric Algorithms for Manufacturing*, Atlanta, Georgia, May 1993. IEEE. .

[9] Karl-Friedrich Böhringer. *Programmable Force Fields for Distributed Manipulation, and Their Implementation Using Microfabricated Actuator Arrays.* PhD thesis, Cornell University, Department of Computer Science, Ithaca, NY 14853, August 1997.

[10] Karl-Friedrich Böhringer, Vivek Bhatt, and Kenneth Y. Goldberg. Sensorless manipulation using transverse vibrations of a plate. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1989 – 1996, Nagoya, Japan, May 1995. .

[11] Karl-Friedrich Böhringer, Vivek Bhatt, Kenneth Y. Goldberg, and Bruce R. Donald. Sensorless manipulation using transverse vibrations of a plate. *Algorithmica*, November 1996. Accepted; to appear.

[12] Karl-Friedrich Böhringer, Russell G. Brown, Bruce R. Donald, James S. Jennings, and Daniela Rus. Distributed robotic manipulation: Experiments in minimalism. In O. Khatib et al., editor, Experimental Robotics IV, *Lecture Notes in Control and Information Sciences 223*, pages 11–25. Springer Verlag, Berlin, 1997. .

[13] Karl-Friedrich Böhringer, Michael Cohn, Ken Goldberg, Roger Howe, and Al Pisano. Electrostatic self-assembly aided by ultrasonic vibration. `http://alpha.ieor.berkeley.edu / karl/MicroSelfAssembly`, 1997.

[14] Karl-Friedrich Böhringer, Michael Cohn, Ken Goldberg, Roger Howe, and Al Pisano. Electrostatic self-assembly aided by ultrasonic vibration. In *AVS 44th National Symposium*, San Jose, CA, October 1997. Accepted for presentation.

[15] Karl-Friedrich Böhringer, Bruce R. Donald, and Dan Halperin. The area bisectors of a polygon and force equilibria in programmable vector fields. In *13th ACM Symposium on Computational Geometry*, Nice, France, June 1997.

[16] Karl-Friedrich Böhringer, Bruce R. Donald, and Dan Halperin. On the area bisectors of a polygon. In *Second CGC Workshop on Computational Geometry*, Durham, NC, October 1997. Extended Abstract.

[17] Karl-Friedrich Böhringer, Bruce R. Donald, and Dan Halperin. On the area bisectors of a polygon. *Discrete and Computational Geometry*, September 1997. Submitted for review.

[18] Karl-Friedrich Böhringer, Bruce R. Donald, and Noel C. MacDonald. Single-crystal silicon actuator arrays for micro manipulation tasks. In *Proc. IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, pages 7–12, San Diego, CA, February 1996. .

[19] Karl-Friedrich Böhringer, Bruce R. Donald, and Noel C. MacDonald. What programmable vector fields can (and cannot) do: Vector field algorithms for MEMS arrays and vibratory parts feeders. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 822–829, Minneapolis, MN, April 1996. .

[20] Karl-Friedrich Böhringer, Bruce R. Donald, and Noel C. MacDonald. Programmable vector fields for distributed manipulation, with applications to MEMS actuator arrays and vibratory parts feeders. *Int. Journal of Robotics Research*, September 1997. Accepted; to appear.

[21] Karl-Friedrich Böhringer, Bruce R. Donald, and Noel C. MacDonald. Single-crystal silicon actuator arrays for micro manipulation tasks. *Journal of Microelectromechanical Systems*, 1997. In preparation.

[22] Karl-Friedrich Böhringer, Bruce R. Donald, and Noel C. MacDonald. Upper and lower bounds for programmable vector fields with applications to MEMS and vibratory plate parts feeders. In Jean-Paul Laumond and Mark Overmars, editors, *Algorithms for Robotic Motion and Manipulation*, pages 255–276. A. K. Peters, Wellesley, MA 02181, 1997. .

[23] Karl-Friedrich Böhringer, Bruce R. Donald, Noel C. MacDonald, Gregory T. A. Kovacs, and John W. Suh. Computational methods for design and control of MEMS micromanipulator arrays. *Computer Science and Engineering*, pages 17 – 29, January – March 1997.

[24] Karl-Friedrich Böhringer, Bruce R. Donald, Robert Mihailovich, and Noel C. MacDonald. A geometric theory of manipulation and control for microfabricated actuator arrays. Technical Report 93–87, Cornell University, Mathematical Sciences Institute, Ithaca, NY 14853, November 1993. .

[25] Karl-Friedrich Böhringer, Bruce R. Donald, Robert Mihailovich, and Noel C. MacDonald. Sensorless manipulation using massively parallel microfabricated actuator arrays. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 826–833, San Diego, CA, May 1994. .

[26] Karl-Friedrich Böhringer, Bruce R. Donald, Robert Mihailovich, and Noel C. MacDonald. A theory of manipulation and control for microfabricated actuator arrays. In *Proc. IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, pages 102–107, Oiso, Japan, January 1994. .

[27] Karl-Friedrich Böhringer, John W. Suh, Bruce R. Donald, and G. T. A. Kovacs. Vector fields for task-level distributed manipulation: Experiments with organic micro actuator arrays. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1779 – 1786, Albuquerque, New Mexico, April 1997.

[28] P. Cheung, A. Berlin, D. Biegelsen, and W. Jackson. Batch fabrication of pneumatic valve arrays by combining mems with printed circuit board technology. In *Proc. Symposium on Micro-Mechanical Systems, ASME International Mechanical Engineering Congress and Exhibition*, pages 16–21, Dallas, TX, November 1997.

[29] Murilo Coutino and Peter Will. The intelligent motion surface: a hardware/software tool of the assembly of meso-scale devices. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Albuquerque, New Mexico, 1997.

[30] Murilo Coutino and Peter Will. Open loop dynamic control of parts moving on an intelligent motion surface along paths composed of straight line segments. In *IEEE COntrol and Decision Conference*, USA, 1997. submitted for review.

[31] Murilo Coutino and Peter Will. Using dynamic vector force fields to manipulate parts on an intelligent motion surface. In *IEEE International Symposium on Automation and Task Planning*, Los Angeles, CA, 1997.

[32] M. Díaz and J. O'Rourke. Ham-sandwich sectioning of polygons. In *Proc. 2nd Canadian Conference on Computational Geometry*, pages 98–101, Ottawa, 1990.

[33] Bruce R. Donald and Dinesh K. Pai. The motion of compliantly connected rigid bodies in contact. *Int. Journal of Robotics Research*, July 1991.

[34] Bruce Randall Donald, James Jennings, and Daniela Rus. Information invariants for distributed manipulation. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 431–459, Wellesley, MA, 1995. A. K. Peters.

[35] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer Verlag, Heidelberg, Germany, 1987.

[36] Michael A. Erdmann. An exploration of nonprehensile two-palm manipulation: Planning and execution. Technical report, Carnegie Mellon University, Pittsburgh, PA, 1996.

[37] J. D. Foley, A. Van Dam, Feiner, and Hughes. *Computer Graphics: Principles and Practice*. Addison Wesley, 2 edition, 1996.

[38] Hiroyuki Fujita. Group work of microactuators. In *International Advanced Robot Program Workshop on Micromachine Technologies and Systems*, pages 24–31, Tokyo, Japan, October 1993.

[39] K. Y. Goldberg. Orienting polygonal parts without sensing. *Algorithmica*, 10(2/3/4):201–225, August/September/October 1993.

[40] Benjamin Joffe. Manipulation and identification of objects by magnetic forces. In *Proc. International Symposium on Magnetic Suspension Technology, NASA Conference Publication 3152 part 2*, pages 617–639, Langley, VA, August 1991.

[41] Lydia Kavraki. Part orientation with programmable vector fields: Two stable equilibria for

most parts. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Albuquerque, New Mexico, April 1997.

[42] Lydia E. Kavraki. On the number of equilibrium placements of mass distributions in elliptic potential fields. Technical Report STAN-CS-TR-95-1559, Department of Computer Science, Stanford University, Stanford, CA 94305, 1995.

[43] Oussama Khatib. Real time obstacle avoidance for manipulators and mobile robots. *Int. Journal of Robotics Research*, 5(1):90–99, Spring 1986.

[44] Daniel E. Koditschek and Elon Rimon. Robot navigation functions on manifolds with boundary. *Advances in Applied Mathematics*, 1988.

[45] Keith Kotay, Daniela Rus, and Marty Vona. The self-reconfigurable robotic molecule. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Belgium, 1998. Submitted for review.

[46] Wenheng Liu and Peter Will. Parts manipulation on an intelligent motion surface. In *Int. Conf. on Intell. Robots & Systems (IROS)*, Pittsburgh, PA, 1995.

[47] Jonathan E. Luntz and William Messner. A distributed control system for flexible materials handling. *IEEE Control Systems*, 17(1), February 1997.

[48] Jonathan E. Luntz, William Messner, and Howie Choset. Parcel manipulation and dynamics with a distributed actuator array: The virtual vehicle. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1541 – 1546, Albuquerque, New Mexico, April 1997.

[49] R. E. Mihailovich and N. C. MacDonald. Dissipation measurements of vacuum-operated single-crystal silicon resonators. *Sensors and Actuators*, 1996.

[50] R. E. Mihailovich, Z. L. Zhang, K. A. Shaw, and N. C. MacDonald. Single-crystal silicon torsional resonators. In *Proc. IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, pages 155–160, Fort Lauderdale, FL, February 1993.

[51] K. S. J. Pister, R. Fearing, and R. Howe. A planar air levitated electrostatic actuator system. In *Proc. IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, pages 67–71, Napa Valley, California, February 1990.

[52] Rama Prasad, Karl-Friedrich Böhringer, and Noel C. MacDonald. Design, fabrication, and characterization of SCS latching snap fasteners for micro assembly. In *Proceedings of the ASME International Mechanical Engineering Congress and Exposition (IMECE)*, San Francisco, California, November 1995. .

[53] M. H. Raibert, J. K. Hodgins, R. R. Playter, and R. P. Ringrose. Animation of legged maneuvers: jumps, somersaults, and gait transitions. *Journal of the Robotics Society of Japan*, 11(3):333–341, 1993.

[54] John Reif and H. Wang. Social potential fields: A distributed behavioral control for autonoomous robots. In K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, editors, *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 431–459. A. K. Peters, Wellesley, MA, 1995.

[55] Dan Reznik, Stan Brown, and John F. Canny. Dynamic simulation as a design tool for a microactuator array. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Albuquerque, NM, April 1997. .

[56] Dan Reznik, John F. Canny, and Ken Y. Goldberg. Analysis of part motion on a longitudinally vibrating plate. In *Int. Conf. on Intell. Robots & Systems (IROS)*, Grenoble, France, September 1997.

[57] Elon Rimon and Dan Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5), October 1992.

[58] Kevin A. Shaw and Noel C. MacDonald. Integrating SCREAM micromechanical devices with integrated circuits. In *Proc. IEEE Workshop on Micro Electro Mechanical Systems (MEMS)*, San Diego, CA, February 1996.

[59] Kevin A. Shaw, Z. Lisa Zhang, and Noel C. MacDonald. SCREAM I: A single mask, single-crystal silicon process for microelectromechanical structures. In *Transducers — Digest Int. Conf. on Solid-State Sensors and Actuators*, Pacifico, Yokohama, Japan, June 1993.

[60] John W. Suh, Steven F. Glander, Robert B. Darling, Christopher W. Storment, and Gregory T. A. Kovacs. Combined organic thermal and electrostatic omnidirectional ciliary microactuator array for object positioning and inspection. In *Proc. Solid State Sensor and Actuator Workshop*, Hilton Head, NC, June 1996.

[61] Z. Lisa Zhang and Noel C. MacDonald. An RIE process for submicron, silicon electromechanical structures. *Journal of Micromechanics and Microengineering*, 2(1):31–38, March 1992.

[62] N. B. Zumel and M. A. Erdmann. Nonprehensile two palm manipulation with non-equilibrium transitions between stable states. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Minneapolis, MN, April 1996.