Spatial/Kinematic Domain and Lattice Computers

John Case[1]
Computer Science Department
103 Smith Hall
University of Delaware
Newark, DE 19716
USA
Email: case@cis.udel.edu
Phone: 1-(302) 831-2711

Dayanand S. Rajan
Computer Science Department
103 Smith Hall
University of Delaware
Newark, DE 19716
USA

Anil M. Shende
Department of Computer Science
Bucknell University
Lewisburg, PA 17837
USA

[1]The research was supported in part by NSF Grant CCR 8713846. We would like to thank Dan Chester for helpful comments on the exposition in an earlier draft.

Abstract

An approach to analogical representation for objects and their motions in space is proposed.

This approach involves lattice computer architectures and associated algorithms and is shown to be abstracted from the behavior of human beings mentally solving spatial/kinematic puzzles. There is also discussion of where in this approach the modeling of human cognition leaves off and the engineering begins.

The possible relevance of the approach to a number of issues in Artificial Intelligence is discussed. These issues include efficiency of sentential versus analogical representations, common sense reasoning, update propagation, learning performance tasks, diagrammatic representations, spatial reasoning, metaphor, human categorization, and pattern recognition.

Lastly there is a discussion of the somewhat related approach involving cellular automata applied to computational physics.

1 Introduction

An important aspect of approaches to solving problems is the *representation scheme* employed for knowledge about the problem. In this paper, we propose a particular *vivid, analogical* (see Selman (1988)) representation for solving problems in the *spatial/kinematic* domain: lattice computer architectures and associated algorithms.

We argue, partly with evidence from the Clinical and Experimental Psychology and Artificial Intelligence literatures, that our approach is superior to approaches using sentential representations (see Sections 2, 5.1, and 5.3 below).

The physical sciences provide problems in the spatial/kinematic domain as do many puzzles and situations ostensibly requiring common sense. Solutions to such problems, at least from the physical sciences, are usually predictions of events in the course of the *unfolding* of a phenomenon under consideration, e.g., predictions of the motions of the participating objects. For spatial puzzle solving, the finding of solutions might involve some trial and error motion of objects in space rather than the tracking of deterministic simulations of physical phenomena. In any case, the knowledge-base for solving spatial problems is the representation of the objects participating, their motions, and the, usually finite, region of space in which the "action" takes place or is to take place.

Most physical phenomena involve the spatial/kinematic domain. Hence, our approach may be well-suited for scientific computing (see Sections 3, 4, and 5.4 below). Scientific computing provides but a subset of the problems we have in mind for our approach (see Section 5.3 below). To what extent our project is cognitive science or engineering is discussed in Sections 5.1 and 5.2 below. We'll give a brief progress report below on how far we've gotten with our lattice computer project in Section 4, but first we motivate it in the next two sections, Sections 2 and 3.

2 Motivations for our Approach

Our lattice computer project (see Case, Rajan and Shende (1990b), Case, Rajan and Shende (1990a), Shende (1991), Case, Rajan and Shende (1991a), Case, Rajan and Shende (1991c), Case, Rajan and Shende (1991b), and Rajan and Shende (1991); also see Section 4 below) was motivated by trying to find a computer architecture naturally fitted to solving problems in the spatial/kinematic domain. For example, we were and are interested in it as an eventual computer vehicle for the learning of non-verbal performance tasks (see Section 5.3 below). Our approach to representing the spatial/kinematic domain originally came out of thinking about

- the work of Nobelist Roger Sperry and others (see, for example, Sperry (1974), Ornstein (1972), Kinsbourne (1982), Springer and Deutsch (1985)) on lateralization of cognitive functions in the human brain and
- the ingenious reaction-time experiments of Shepard (with students and reported, for example, in Shepard and Metzler (1971), Shepard and Feng (1972), and Shepard and Cooper (1982)) and Kosslyn (also with students and reported in Kosslyn (1980) and Kosslyn (1983)) on human subjects *mentally* solving *spatial/kinematic* puzzles

(see also Section 5.1 below).

The work on lateralization of cognitive functions in the human brain shows us that the half of the brain (right half) that is generally good at spatial puzzles is not good at verbal puzzles and vice-versa. It occurred to us, then, to look to *non-verbal* representation schemes for the computer solution of spatial/kinematic problems. Kinsbourne (1982) posits, among other things, that the right brain hemisphere provides a *spatial framework*.

Shepard and Metzler (1971) showed that the time it takes people to *mentally* match pictures of two congruent, complex 3-dimensional objects, one rotated with respect to the other, is *linear* in the angle between them. In general, the reaction-time experiments on "mental-imagery" are consistent with human beings employing an *analogical* (see also Funt (1980), Rumelhart and Norman (1974), Sloman (1971), Sloman (1985), Selman (1988), Forbus (1981), Forbus (1983), Hinton (1979), Brooks (1981), McDermott and Davis (1984), and Larkin and Simon (1987)) representation scheme for spatial/kinematic problems. It is as if we project a mental analog of space and mentally move analogs of objects around in it, at least in simple cases, with timing approximately linear in the real time it would take to move the corresponding real objects similarly in real space. At least that is a more parsimonious model (consistent with the data) of human representation for spatial/kinematic domains than assuming that people subconsciously directly solve systems of equations, multiply matrices, or run automatic theorem provers and then subconsciously make the reaction-time experiments come out suggesting a more analogical approach.

As will be seen in Section 4, our lattice computer architectures *are* spatial frameworks discretely representing space, and, as will be noted, our associated algorithms (so far developed) carry out *discrete but analogical* simulations of motion in space *approximately in real-time*: 'approximately in real-time' means, here, as close as possible to linear in real-time.

3 Inherent Parallelism and Ergonomics

It occurred to us that in the physical movement of a solid object, each particle of the object moves *in parallel* with each other particle; the particles do not move first one, then another, then another, ... It appears, then, that most sentential and mathematical representations do not, and perhaps cannot, directly make use of *this* inherent parallelism in spatial phenomena. To be sure, many *parallel* numerical algorithms which can simulate the motion of objects in space too. Kung and Leiserson's algorithm for matrix multiplication in a hexagonal mesh (Kung and Leiserson (1978)) and Kosaraju and Flynn's earlier one (Flynn and Kosaraju (1976)) for square meshes are, *in large part*, conceptualized in terms of the *motion of numerical data in space.* It just seems to us to be more natural and efficient to try to exploit the inherent parallelism of individual particle motion directly, and that is how our project is proceeding.

It also occurred to us that, if we could represent the inherent parallelism in particle motion directly, this should eventually lead to an ergonomic soundness for the design phase of associated algorithms. This contrasts with most current approaches to parallel computation which require a lot of pain in algorithm design to exploit that parallelism Shende (1991).

The point about ergonomic soundness is justified, in part, by the considerable literature in experimental psychology (discussed above briefly in Section 2 and further discussed in Section 5.1 below; see Shepard and Cooper (1982), Kosslyn (1980), Kosslyn (1983), and

Rumelhart and Norman (1974)) suggesting that *human* mental imagery is analogical and approximately real-time, not analytical. For example, the time it takes people to mentally rotate 3-dimensional objects is *linear* in the angle through which the objects are to be rotated (see Shepard and Metzler (1971) and Shepard and Cooper (1982)). Of course the time it takes to *analytically* simulate rotation by standard matrix operations is *not* linear in the angle.

4 Our Approach and Lattice Computers

Our approach can be roughly described as one *representing space by space and time by time*. We will first explain our approach by an example. Although this example does not quite accurately present what we really do, it provides a good way to begin explaining our approach. Additionally, it helps in precipitating some issues that need to be resolved.

Consider the problem of simulating the uniform motion of a 2-dimensional object, $\boldsymbol{O}$, in a finite, convex region, $\boldsymbol{R}$, of a copy of euclidean 2-space (we are actually much more interested in dimensions three and above). For simplicity, assume that $\boldsymbol{O}$ is the only object in $\boldsymbol{R}$.

We *choose* a finite set of points from $\boldsymbol{R}$. We place identical, synchronized computer processors at each of the points in this set, and connect some of these processors by bi-directional communication channels with delay time proportional to length. Our choices could produce the mesh-work of computer processors representing $\boldsymbol{R}$ shown in Fig. 1(a). The circles (filled and unfilled) stand for the processors, and the solid lines for connections. Note that we did make some (possibly peculiar) choices to arrive at this particular mesh-works—we will refer to this matter below. Imagine that somehow each of the processors in this mesh-works is programmed to "behave" like the point of $\boldsymbol{R}$ it is placed on. So, we would have some processors programmed to behave like empty space (these are the unfilled ones in Fig. 1(a)), and others (the filled ones in Fig. 1(a)) programmed to behave as particles of $\boldsymbol{O}$. Motion is represented, then, by the synchronized passage of messages, across direct communication channels, from one processor to another. Imagine that a processor that represents a particle of $\boldsymbol{O}$ instructs a directly connected processor in the direction of motion, to start representing that particle. In the example shown in Fig. 1(a) each neighboring processor in the direction of motion happens to be a constant length away from the particles of $\boldsymbol{O}$, and, since all the processors in the mesh-works execute in synchrony (or in parallel), after one message passing, $\boldsymbol{O}$ would effectively have moved in the direction of motion. The arrows in Fig. 1(a) indicate the direction of motion of $\boldsymbol{O}$. Fig. 1(b) shows the state of the mesh-works, as far as what each processor represents, after one message passing. Repeating this process over and over again, *if it were possible*, would result in the mesh-works simulating the uniform motion of $\boldsymbol{O}$ in $\boldsymbol{R}$; moreover, the simulation time could come out *linearly proportional* to the actual time $\boldsymbol{O}$ would take to move in $\boldsymbol{R}$.

Generalization of this description to higher dimensions should be fairly clear, but it should also be clear that our choice of discrete points and connections to represent $\boldsymbol{R}$ is fairly higgledy-piggledy except for enabling us to cleanly present the example, one step only motion depicted in Fig. 1. About this one step motion example, the reader may even suspect (correctly it turns out) that we "cheated". Early on in our project we worried a lot about *which* set of points and interconnections to use to represent a region and, for that choice, how to deal with motions not so cleanly representable as the example in Fig. 1. For example, for a mesh-works of points arranged and connected in squares, but without diagonal connections, we had to deal with the question of how to handle diagonal motion at roughly the same *speed* as motion along the edge of a square. The example from Fig. 1 doesn't deal with this and related problems, but our algorithms, discussed briefly below, do.

First we'll discuss how we've actually dealt with the selection of points and interconnections (see also Case, Rajan and Shende (1990a) and Case, Rajan and Shende (1991a)). Then we'll say a little about our algorithms to date featured in Shende (1991), Case, Rajan and Shende (1991c) and Case, Rajan and Shende (1991b).

Of course, if real physical space is discrete (instead of continuous like euclidean spaces), it is not clear whether the points and interconnections of real space are regular, and if so with what regularity, whether it is irregular, whether it changes with time, ... Minsky (1982), Fredkin. It could be rather a mess. Also, if real space is discrete, its resolution is likely to be much finer than we could hope to precisely represent with a mesh-works computer for macroscopic level phenomena; furthermore, real space may have no minimal distance between points. We simply adopted for the points and interconnections in our computer meshes very regular schemes which have *very nice properties*, including, of course, a minimal distance between points. Unlike for the approach of Funt (Funt (1980), Rumelhart and Norman (1974)) and for the numerical finite element method (see, for example, Chandrupatla and Belegundu (1991)), we decided that we didn't want our discrete version of a region of euclidean space to have its points having thiner and thiner density as one moves outward from a area of high interest in the region. Very much unlike Funt (see the further discussion in Section 5.3 below), we chose to have our arrangements of points to be (for the infinite extension of a bounded region) invariant under translations. Roughly, these design decisions of ours are provably mathematically equivalent (Case, Rajan and Shende (1990a) and Case, Rajan and Shende (1991a)) to having the set of points which represent all of a copy of euclidean n-space be an *n-dimensional lattice* Conway and N. J. A. Sloane (1993), Gruber and Lekkerkerker (1987). An n-dimensional *lattice* is a set of points, from some copy of euclidean n-space, providing a discrete variant of an n-dimensional vector space (see, for example, Herstein (1964)) in euclidean n-space in which the scalar multipliers for magnifying vectors are restricted to being integers. For example, $\mathbf{Z}^n$, the set of n-tuples of integers, and $\mathbf{A}_n$, the set of $(n+1)$-tuples of integers that add up to 0, are both n-dimensional lattices Conway and N. J. A. Sloane (1993). For $\mathbf{A}_n$, the associated copy of euclidean n-space is the set of $(n+1)$-tuples of *reals* that add to 0. Clearly, for example, the set of 3-tuples of *reals* that add to 0 is just a plane, a copy of 2-dimensional space.

In Case, Rajan and Shende (1990a) and Case, Rajan and Shende (1991a) we define an n-dimensional lattice to be *regular* iff it can be generated from integer linear combinations of some set $\boldsymbol{B}$ of n vectors which are minimal length for the lattice and such that any two distinct vectors in $\boldsymbol{B}$ are a constant angle apart. Regularity is a powerful symmetry condition to the effect that with respect to some set of n generating vectors one dimension is not to be preferred to another; it also considerably simplifies vector computations. Both $\mathbf{Z}^n$ and $\mathbf{A}_n$ are regular n-dimensional lattices. $\mathbf{D}_n$, the set of all n-tuples of integers which sum to an even number, (Conway and N. J. A. Sloane (1993)) is an n-dimensional lattice generated by its set of vectors of minimal length, but, for $n > 3$, is not regular.

We associate any bounded region of a copy of euclidean n-space and an n-dimensional regular lattice embedded in that copy of n-space with a meshworks computer having identical, synchronized processors on each lattice point in the region and processors immediately connected (by identical, bi-directional communication channels) iff they are the *lattice minimal distance* apart. We call such a mesh computer, a *lattice computer*. We represent bounded regions of euclidean n-space by particular lattice computers.

We prove in Case, Rajan and Shende (1990a), Case, Rajan and Shende (1991a) that $\mathbf{A}_n$ is the unique (up to angle-preserving isomorphism) regular n-dimensional lattice with the *maximum* number of nearest-neighbors among the n-dimensional *regular* lattices. Each point in $\mathbf{A}_n$ has $n(n+1)$ nearest-neighbors; each point in $\mathbf{Z}^n$ has but $2n$ nearest-neighbors. We chose, then, to base our lattice computer (Case, Rajan and Shende (1990b)) on the lattice $\mathbf{A}_n$. Case, Rajan and Shende (1990b) also contains a top level design for the individual processors and a feasible scheme for embedding our n-dimensional lattice computers, $n > 3$, in real physical 3-space. (We mean feasible with respect to wire length and crowding Vitányi (1988). We expect that it would be too expensive to build one of our lattice computers for dimension much bigger than 3 if there are to be on the order of 10^2 processors along each of at least 3 dimensions.) For $n > 3$, an embedding into physical 3-space of a lattice computer based on $\mathbf{A}_n$ would involve having to "bend" some connections; however, simulated objects traveling in a machine so embedded would, of course, be insensitive to the bends.

Fig. 2 shows a picture (with opaque processors) of a lattice computer based on $\mathbf{A}_3$ representing a small parallelepiped region of euclidean 3-space.

In Shende (1991), Case, Rajan and Shende (1991c) and Case, Rajan and Shende (1991b) we present algorithms associated with the euclidean transformations. The algorithms include discrete versions of constant speed particle translation, non-dissipating spherical wave-front motion and circular revolution of a particle about a point, all in n-space and for a wide class of n-dimensional lattice computers, including ours. (We recently discovered Kochevar (1991) which presents an algorithm for the energy *dissipation* of light in 3-dimensional cubic lattice computers—essentially lattice computers based on $\mathbf{Z}^3$.) The lattice computers for which our algorithms work are based on a class of lattices described at the end of the fourth paragraph below this one. These lattices are characterized by Rajan and Shende Rajan and Shende (1991), Conway and N. J. A. Sloane (1993) as the well-known root lattices (possibly rescaled).

Our discrete, constant speed particle translation and (circular) revolution algorithms are instances of an algorithm schemata, CURVE Shende (1991) and Case, Rajan and Shende (1991b). CURVE analogically simulates the constant speed motion of a particle moving along any one of a large class of curves. Straight lines and circles are among this class of curves, and the corresponding two instances of the schemata CURVE simulate, approximately in real–time, the constant speed translation and the constant angular speed circular revolution of a particle, respectively. Processors running algorithm schemata CURVE are in one of two states: *active* or *passive*. Active processors track the motion of the particle being simulated. Unlike, say, spherical wavefronts, where every processor is eventually on some simulated wavefront, curves could pass through the empty space in between processors, not hitting any processor for long stretches. Hence, we have each processor represent a whole region of n-space around it. Suppose $\boldsymbol{X}$ is a point in an n-dimensional lattice. Consider the points in n-space that are at least as close to $\boldsymbol{X}$ as to any other point in the lattice. This set of points forms a convex, n-dimensional body around $\boldsymbol{X}$ called the *Voronoi cell around* $\boldsymbol{X}$ Conway and N. J. A. Sloane (1993) and Aurenhammer (1990). Voronoi cells have $(n-1)$-dimensional faces, the Voronoi cells around individual lattice points are congruent to one another, and they tessellate n-space.

Figure 3 shows a lattice computer based on $\mathbf{A}_2$, and a curve $\boldsymbol{C}$ in 2-space traced by a moving particle starting at the point/processor, $\boldsymbol{S}$, in the lattice computer. Figure 3 shows the Voronoi cell (with a dashed-line boundary) around the point/processor labeled $\boldsymbol{P}$ in the lattice computer based on $\mathbf{A}_2$. In algorithm schemata CURVE, each point/processor is "responsible" for points in its Voronoi cell. Algorithm schemata CURVE actually treats the curve $\boldsymbol{C}$ as a sequence of small, equal length straight line segments. The *length* s of these segments is, essentially, a parameter to CURVE Shende (1991). To minimize the number of different Voronoi cells a segment of length s can intersect, s should be chosen less than an upper bound that depends on the lattice; however, in practice, one chooses s much smaller than this upper bound. For $\mathbf{A}_n$, the upper bound is one-half the lattice minimal distance.

Henceforth, by $\boldsymbol{C}$ we will mean the corresponding curve made up of the tiny straight line segments. As the particle moves along the curve $\boldsymbol{C}$, it enters and exits Voronoi cells around some of the points/processors in the lattice computer. Algorithm schemata CURVE ensures that the point/processor $\boldsymbol{S}$ is initially active at simulation time 0, and, afterward, that each point/processor $\boldsymbol{P}$ is active exactly in a duration of simulation time uniformly approximately linear in the real–time duration that the particle spends inside, or on the boundary of, the Voronoi cell around $\boldsymbol{P}$. Each point/processor $\boldsymbol{P}$ receives a message stating approximately when and where the particle entered the Voronoi cell surrounding it and any additional information it needs to compute the motion of the particle within that Voronoi cell. $\boldsymbol{P}$ then computes approximately when and where the particle will leave the Voronoi cell and, at the appropriate time, sends a message to each neighboring processor into whose Voronoi cell the particle might enter next. Again, a point/processor becomes active if and only if the particle actually enters the Voronoi cell around it.

It is beyond the scope of the present paper to present in detail any of our algorithms, but intuitively they perform delicate calculations of (i) the *time* a processor must wait after receiving a message before it can pass a message to a nearest-neighbor processor and (ii) calculations of which nearest-neighbor(s) to send which messages based on which nearest-neighbor(s) sent which messages. Being able to work with integer vector arithmetic is a big help. Recently our algorithms have been extended to the cases of N non-interacting particles or waves in Case, Chitoor, Rajan and Shende (1993). The class of lattices for which all our algorithms work are those whose Vornoi cells have inscribed n-spheres (see Rajan and Shende (1991) and Conway and N. J. A. Sloane (1993)).

Essentially these algorithms discretely simulate the corresponding continuous motions, with simulations approximately real-time, and where the error in the approximation can in most cases be bounded arbitrarily small. Unlike round-off errors in numerical calculations, the errors here are controlled. We should note that we pay a small price regarding circular motion for the design decision of invariance under translations. Circular motion provably (Shende (1991)) cannot be as perfect as straight line motion. For circular motion in a lattice computer, we define an *angular resolution* which depends on s and the euclidean diameter of the lattice computer. Our algorithm for circular motion requires us to work with an angle that has rational sines and cosines and that *approximates* the resolution angle. This approximation results in a slight drift from perfect constant angular speed. However, the drift can be made so small that it cannot be seen within the angular resolution of the lattice computer until after as many thousands of revolutions as we like. This control of the drift is achieved by tiny and feasible adjustments in the approximating angle (Shende (1991)). Funt's spatial framework (Funt (1980) and Rumelhart and Norman (1974)) handles circular motion (about one axis) better than translations (see the discussion in Section 5.3 below).

Some readers may think we are wasting processors in our simulations: those simulating empty space at any time seem to be doing nothing at that time. We don't look at it that way: they *are* properly simulating empty space. We do *not* subscribe to a parallel computing philosophy that says that almost all processors should be *actively* computing something at any time. That attitude is correct for most other current approaches to parallel computing, but leads to a lot of pain (see Section 3) in the algorithm design phase for those approaches.

We are in the process of extending our algorithms to handle interacting wavefronts, magnification of figures, accelerated motion, and multiple, joined or otherwise interacting particles.

Furthermore, many things are left to decide about hardware and I/O before we can build a prototype of our lattice computers. We expect that the first useful applications will involve discrete analog scientific computing. We hope to develop algorithms to solve, say, classical physics problems, especially fluid flow problems. (See further discussion in Section 5.4 below.) *Thinking Machines* (see Thinking Machines (1986), Salem and Wolfram (1986)) successfully employed their Connection Machine to *analogically* simulate 2-dimensional fluid flow problems in a similar fashion. (*Thinking Machines*'s video tape of this is impressive.) They *simulated* a hexagonal 2-dimensional mesh with their Connection Machine, essentially an SIMD hypercube that has an interconnection network (Siegel (1985)) and a routing algorithm to *simulate* arbitrary connection schemes. Simulation on a Connection Machine likely can't be carried out approximately real-time because of unpredictable delays in the interconnection network.

We expect our lattice computers will be useful in problems involving real-time control.

5 Discussion

5.1 Human Non-Verbal Cognition and Perception

Many dichotomies have been proposed to capture conceptually the essence of the intriguing cognitive differences between the left and right hemispheres of the human brain Springer and Deutsch (1985), Ornstein (1972), Kinsbourne (1982). We shall not go into all these here. Suffice it to say that some of them, e.g., rational vs intuitive thought, while suggestive, are seemingly a little too operationally ill-defined to admit of testing in experimental and/or clinical settings. Others such as basic language competence vs the simplest spatial skills are much more amenable to clear testing. For example, most human adults (especially right-handed males, the most cognitively lateralized) who suffer *extensive* damage to their left-brain hemispheres (through stroke or other trauma) are thereafter unable to understand or communicate in written or spoken language, but they retain competence to more or less correctly copy simple line drawings of cubes, houses, etc. (they may leave out some details or simplify); furthermore, they retain roughly their prior competence level to solve simple spatial puzzles such as Koh Block Designs (Springer and Deutsch (1985), Gardner (1975), Gardner (1982), and Ornstein (1972)); just the opposite occurs with extensive damage to the right hemisphere: most language functions will remain intact, but the spatial skills will be severely impaired. Interestingly enough, on tasks such as copying a simple line drawing of a house, those with right-hemisphere damage may copy all the details (windows, corners, and such like), but display these details without connections to one another, strewn all over the paper: the resultant drawings will not resemble a house or capture anything of its 3-dimensional character. (In the case of right-hemisphere damage there may also be neglect of anything in the left visual field, but this curiosity is not so relevant to the points we want to make.) Kinsbourne (1982) posits that the left hemisphere handles details and the orderly sequence in which they are to be dealt with; whereas, (as we noted in Section 2 above) the right hemisphere provides a *spatial framework* in which to make entries.

In general it seems that both brain hemispheres in normals must contribute to non-verbal, simple spatial puzzle solving, although, if one hemisphere cannot or is not allowed to participate (as with brain damaged subjects and the split brain patients studied by Sperry and his students), then the right hemisphere is superior. For example, there is right brain superiority at mental rotation tasks (see Springer and Deutsch (1985)). Possibly, when one hemisphere alone must work on *most* spatial tasks, the maintenance of a spatial framework is *much* more important than being extremely careful with details and the orderly sequencing of steps.[4] (We said '*most* spatial tasks' because those spatial tasks that are employed clinically do seem to be prototypical of spatial tasks and intuitively seem to require a spatial framework. One can suspect that for the "spatial" task of connectivity recognition, studied by Minsky and Papert (1969), there would not be right-hemisphere superiority. Minsky and Papert explicitly employed this task to show the severe limitations of simple "holistic" computers (perceptrons), but they showed that machines with cycloptic tunnel vision (Turing machines) could easily handle it.)

A sense in which we are modeling right brain hemisphere computation with lattice computers is that, ostensibly like right brain hemispheres, a lattice computer provides a spatial framework. For other aspects of human spatial cognition, it is not so clear what the contribution of each hemisphere is. We're more interested in modeling spatial cognition than in just modeling the right hemisphere's contribution. Naturally we are talking about high level modeling, not low level, neural net modeling. There is no reason to believe that a portion of the wetware of the brain's architecture *literally* corresponds to, say, a 3-dimensional lattice computer. The cortex of the cerebrum itself, including the visual cortex, is essentially a 2-dimensional surface (see Cook (1986)). We would expect that the reduction, to the neural net level, of high level mental imagery of 3-dimensional transformations is not quite so simplistically straightforward. (There is further discussion of levels of modeling in science in this section.)

Nonetheless, the experimental human reaction time results for a wide variety of tasks apparently requiring dynamic mental imagery, tasks including *mentally* checking congruence of pairs of complex 3-dimensional shapes (Shepard and Metzler (1971) and Rumelhart and Norman (1974)), the *mental* solving of paper folding puzzles (Shepard and Feng (1972) and Rumelhart and Norman (1974)), etc., invariably produce times linear in the real time to physically carry out actual real world rotation of shapes for matching, the folding of paper shapes, etc. (Shepard and Cooper (1982), Kosslyn (1980), Kosslyn (1983), and Rumelhart and Norman (1974))! Since our simulations so far are approximately real-time, this is an important indicator of the correctness of using our lattice computers to model *at a high level* human non-verbal spatial cognition. N.B. We distinguish, of course, between non-verbal *cognition* and non-verbal *perception*; solving spatial puzzles involves cognition, at least low-level vision does not. However, there does seem to be important sharing of brain "machinery" (at *some* level) between vision and production (if not manipulation) of mental images. Many ingenious experiments show that the same unusual perceptual effects (which effects are unlikely to be known to the subjects in advance) occur with both real images *and imagined ones* (see Finke and Schmidt (1977), Shepard (1978), Finke (1980), Block (1981), and Kosslyn (1983)).

There are various known topographic maps in the cerebral cortex, 2-dimensional patterns on the essentially 2-dimensional cortex (Cook (1986)), *some* of which patterns are roughly (with topological and other distortion) pictorially isomorphic to what they represent. It is unknown to what extent 2-dimensional *mental* imagery involving 2-dimensional transformations (only) are mirrored pictorially in corresponding 2-dimensional motions of various roughly pictorially corresponding 2-dimensional retinotopic (or other topographic) maps.[1] In the terminology of Shepard and Chipman (1970) such an isomorphism of mental rotation and corresponding brain representation is called *first-order.* It very unlikely to exist for 3-dimensional mental imagery if the brain representations are to be confined to the essentially 2-dimensional sheet of brain cortex. Thanks to the reaction time experiments involving 3-dimensional imagery, there must, nonetheless, be a kind of strong, but less literal, isomorphism between the transformations of mental imagery and the corresponding (high level) brain representations (but see three paragraphs below). In the terminology of Shepard and Chipman (1970) this sort of strong isomorphism that does exist is called *second-order.* Our lattice computer representations of the motions of objects in 3-dimensional space are, then, roughly second-order isomorphic to the corresponding acts of human spatial cognition, but first-order isomorphic to the actual motions of objects in space.

It might be argued that, since human visual images, real and imagined, are continuous, our lattice computers automatically fail to model them. However, there is a lower limit to the resolution of human visual images: they actually have a graininess to them (see Kosslyn (1983)); continuity of images is just a, no doubt extremely useful, illusion. (N.B. Many writers in computer science, cognitive science, philosophy, and psychology seem to think that the operation of *all* analog devices must be continuous, not discrete. Even analog computers which have already been built most likely do not quite operate continuously: electrons are discrete. If real physical space is actually discrete, no analog device can be constructed which operates continuously. There is a fairly widespread, but not quite as widespread, misconception that pictorial manipulation cannot be done by *digital* computers, presumably because pictures are believed to be continuous. Of course every *real* picture we know of is grainy at some level of resolution. Algorithmic processing of grainy pictures does not go beyond Turing machine computability—see, for example, Case (1987).)

The simpler experimental human reaction time results for mental manipulation of objects in space, which yield times linear in the real time to carry out such manipulations on real objects, apply to objects of up to 3-dimensions. There was probably no evolutionary pressure for humans to develop manipulative imagery systems with a corresponding mental spatial framework of higher dimension than three. Our lattice computers can, in principle, be constructed for any finite dimension, and, in this respect too, they go beyond a human limitation.

Kosslyn (1983) posits the existence of a mental space in which people carry out mental spatial manipulations and which, in Shepard and Chipman's terminology (Shepard and Chipman (1970)), is second-order isomorphic to an essentially discretized, finite, connected, convex region of two or three dimensional euclidean space. He reports many clever experiments designed to determine the nature of this mental space. Some of the cleverness was to make sure that subjects were not (perhaps subconsciously) going along with what the experimenter wanted. Kosslyn (1983) describes a series of experiments by various of his students (Pinker, Finke, and Shwartz) to investigate whether the mental space is two or three dimensional. He suggests that, while the mental space has many characteristics of a 3-dimensional space, it is a 2-dimensional medium which can be processed so that, except on especially tricky tasks, it appears to be 3-dimensional. We suspect the final word is not in on this, but, again, we are not interested in closely modeling unfortunate human limitations. We are reminded, though, of Marr's notion for vision (not imagery) of a $2\frac{1}{2}$-dimensional representation of the 3-dimensional world (Marr (1982)). Perhaps there is a correspondence between this $2\frac{1}{2}$-dimensional representation for *vision* and Kosslyn's more-than-2-but-not-quite-3-dimensional space for *mental imagery*. It would be intriguing for appropriate experimentalists to work out a correspondence.

It has been noted that there are severe memory limitations on human mental imagery and that these images fall apart if manipulated too much (Kosslyn (1983)), but, again, we're not interested in modeling human shortcomings.

In all honesty we should note that there is a philosophical debate about whether human mental imagery might be a mere epiphenomenon (Block (1981)). The problem is whether mental imagery is as inessential to those acts of cognition which we believe involve mental imagery as the flashing lights on a computer console are to the computations inside the machine. One can argue that since imagistic transformations are carried out in time linear in the real time to carry out the same operations on real objects, and, since (as noted above) image production shares some unusual "perceptual" effects with vision, imagery is not at all a mere epiphenomenon. Consider the following mental puzzle.

Many socks have an emblem on either the right or left side, but not on both. One sock of a pair will have the emblem on the right, the other will have it on the left. Suppose we try to put on such socks so that these emblems will not face one another when we are standing with our legs together. For example, then, when seated, if we cross one leg over the other, the emblem will not show on the top of the top leg. Suppose one of these socks is turned inside out, but we put it on before we discover that. After we discover the problem, perhaps by noticing the emblem is a little indistinct or that the toe stitching is protruding, we remove the sock, turn it right side out and put it on again. Here's the mental puzzle. When we put the sock on the second time, do we have to put it on the same or a different foot? No fair guessing, trying an experiment with a real sock, remembering the answer stored from having tried that or a related experiment in the past, or jumping to conclusions (perhaps correct) from prior knowledge stored about the mathematics of reflections; furthermore, the solution must come from some form of conscious thought which not only gets the right answer, but justifies it: no fair just having the right answer come to you—that's too hard to tell from guessing.

Some people solve this puzzle quickly and according to the (somewhat cumbersome) rules, but we really doubt if anyone can do that without manipulating a mental image. It seems very unlikely a solution could be obtained, both quickly and, say, in time linear in the real time to try it out with a real sock, by means of analytic manipulations of equations or derivations from *non ad hoc* axioms. Hence, if imagery is a mere epiphenomenon and *all* of human cognition is really (mostly subconscious) manipulation of equations or wffs, people would be very unlikely to solve such puzzles *the first time they encounter them* with the *timing* they do.

Incidentally, consider how (if at all) you checked the mutual consistency of these two sentences from above: "Suppose we try to put on such socks so that these emblems will not face one another when we are standing with our legs together. For example, then, when seated, if we cross one leg over the other, the emblem will not show on the top of the top leg." We bet you did it, if you did it, by consulting a real leg *or a mental image of one*.

Some people are fairly conservative as to what constitutes a proper model of (some fragment of) human cognition. They demand something like a complete reduction to neural nets and/or neural biochemistry; for them, nothing short of that will do. In other parts of science, complex phenomena are modeled *with good predictive power* at various levels. For example, no one would seriously consider modeling in biology how predator-prey population sizes oscillate in dynamic or unstable equilibrium on the basis of reduction to the biochemistry (or worse yet, quantum mechanics) of the separate organisms involved. That example is, of course, rather extreme, but is meant to make a point strongly: scientific modeling of complex phenomena can be done at less than a most reduced level, and sometimes it is good sense to do so.

There has been, as there should, work on connectionistic or neural net modeling of mental imagery, including mental rotation (see Baron (1985b) and Baron (1985a)). Baron notes in Baron (1985b) and Baron (1985a) that his modeling does not, though, naturally lead to computation times linear in the time for the real world events imaged to unfold.

We should note that Kosslyn's modeling is high level, and he has developed a computer model (see Kosslyn (1983)), but, since he employs standard analytic representations, his model does not run in time linear in the real time for the imaged events to unfold.

Our model should suggest further work for experimentalists to tease out the exact nature of the left and right hemisphere's contribution to human non-verbal cognition. Of course there is work to be done (by us for the lattice computers, and by psychologists for human cognition) on how to model the (for humans, many times subconscious) decision making regarding which images to have and which spatial transformations to perform.

Our work may shed some light on possible functions of the *corpus callosum* (see Cook (1986)), the most massive bundle of nerve fibers connecting the cerebral hemispheres in placental mammals. (Marsupials, for example, don't have this massive connecting bundle; they have analogs of the other, much smaller, connecting bundles.) The *calloso-bulbar index* of a placental mammal is (by definition in Anthony (1938) and Blinkov and Glezer (1968) and in Cook (1986), Page 9) the ratio of the cross-sectional area of nerve fibers in its corpus callosum to that of the ascending and descending nerve fibers in its brain stem. One would expect these cross-sectional areas to be approximately proportional to the number of fibers cut by each cross-section. Intuitively, the number of nerve fibers in the corpus callosum is a rough measure of the brain power devoted to brain cross-talk; the number passing through the brain stem is a rough measure of the brain power for dealing with the body's external (and internal) effectors and sensors; now, *perhaps* the ratio *roughly* quantifies part of the brain power given over to pure thought, or at least part of "thought" not directly related to immediate interaction with the outside world (or body organs). In any case, in humans the calloso-bulbar index is 3.12, in chimpanzees it is a mere 1.79, for polar bears it is 1.06, and for the hippopotamus it is 0.56 (see Blinkov and Glezer (1968) and Cook (1986), Page 9).

Here is a possible connection between our work and the corpus callosum. We will want a host computer to help with I/O for our lattice computers. That host could be a powerful, but more standard computing engine in its own right, and the connection between the two machines can be reconceptualized so that each machine is construed as a host to the other: perhaps each brain hemisphere functions as a host computer to the other with the corpus callosum playing the role of principal bundle of host connecting wires.

Regarding cognition in physics we note the following. Analogical thinking is natural for physics and engineering. For example, visual and kinesthetic imagery, not verbal thinking, was the cognitive forte of physicist Albert Einstein (see Schilpp (1949)) and inventor Nikola Tesla (see O'Neil (1981)). Maxwell's early model of electromagnetic phenomena was quite mechanical/visual complete with rotating cylinders and idle wheels (see Nivan (1961)). Once *a lot* more experience is had with our lattice computers, they may become a tool for thinking out new physics and engineering too. It is probably no accident that the words 'imagine' and 'image' have a common root.

5.2 Airplanes Don't Make Nests

For good reasons the Wright brothers didn't design feathered, wing-flapping airplanes. It is one thing to understand how *birds* fly, quite another to *abstract* (and adapt) salient features of that process as the basis for designing an artifact in which people can fly. The abstractions of air-foil based lift and propulsion are principal salient features. For air-foil, the shape of the wing and how a fluid made of tiny particles flows over it are what are important. To *completely* understand *bird* flight, it would no doubt be required to include a quantum mechanical explanation of the biochemistry of both feathers and how energy is generated for flapping. To *understand and apply* the *abstraction* of flight is quite another thing.

We want to understand and apply, with improvements, an abstraction from human spatial cognition.

Organisms (including humans) were mostly "designed" by mindless evolutionary forces to simultaneously satisfy lots of engineering-like constraints (the partial exceptions to the mindless part are those organisms purposefully bred by people to have certain characteristics). Here are some of the constraints. The organism must descend from close ancestors that are not too unlike it, it must survive to successfully reproduce (a big constraint), if it has a brain at all, it must have a brain it can easily carry around and feed, this brain must be good at many tasks especially those related to basic needs, ... Birds, but not airplanes, have to build nests, find mates, etc. Artifacts do not have to satisfy the same or (fortunately) as many constraints as biological organisms. Many times, then, an artifact analog of some biological process can be designed to considerably turn up the gain on some special talent relative to the original biological case. Airplanes fly faster than birds. Ordinary computers do arithmetic faster than people.

The human cerebral cortex is a 2-dimensional sheet, in part, because it is for non-human ancestors quite a ways back. Perhaps originally the cerebral cortex was 2-dimensional, also in part, because the body's principal, *external* sensory surfaces (tactile and visual) are 2-dimensional. In any case, we are not so constrained in the dimensionality of our lattice computers, our lattice computers are not necessarily as limited in memory capacity and resolution for images as people are, and we are not required to make the lattice computers out of slow neural nets.

However, for example, both bird and airplane design are constrained to handle a common problem, the abstraction of propulsion; but, as is well known, they needn't handle it the same way. No doubt biological brain design is somewhat constrained by the need to cool circuits and have access for "wiring". We have to deal with such problems and other feasibility issues for our lattice computers.

5.3 Artificial Intelligence

Sloman (1971) pointed out several disadvantages, for certain problem domains, of *Fregean* representations, i.e., ones that use predicate calculus, or some other sentential form of representation, over *analogical* representations. (Henceforth, we will use the terms Fregean, sentential, propositional, and declarative (see Genesereth and Nilsson (1987)) interchangeably.) Spatial problems are from such problem domains, and hence these disadvantages apply. We summarize below the disadvantages of Fregean representations and the advantages of analogical representations, partly as seen by Sloman, partly as seen by others including ourselves.

Interpretation of or extracting information from a Fregean representation may involve complex procedures. In a Fregean system, although the structure of the expressive medium need not constrain the variety of structures of configurations which can be so expressed (represented), one needs procedures for dealing with and accessing *efficiently* the representation for any problem domain. Furthermore, changes made to the description, especially in the case of spatial phenomena, are not so usefully/efficiently related to the changes in the structure of the represented configuration. The motion of objects in space can create a severe update propagation problem for a Fregean representation scheme, but not for our scheme.

On the other hand, translations from and into spatio-temporal configurations will, likely, involve less complex, efficient procedures if the internal representations use a medium *analogous* to space-time. In analogical representations, small changes in the representation (syntactic) are likely to correspond to small changes in what is represented (semantic). Changes in a certain direction or dimension in the representation correspond, directly, to similarly related changes in the represented configuration. In an analogical representation, all relationships involving any certain object can be obtained "locally". Here we are especially concerned with motion in the spatial domain. We are concerned that both the *timing* of motion and the spatial placement of objects be strongly vividly analogical. We are reminded here (and in Section 5.1 above) of Shepard's "definition" (Selman (1988) and Shepard and Cooper (1982)) of an *analogical process* which seems to capture the essence of this timing requirement: an *analogical process* is one by which the intermediate internal states have a natural one-to-one correspondence to appropriate intermediate states in the external world. We would add that the one-to-one correspondence be computable approximately real-time too as for our strongly vividly analogical representations.

Since we are dealing with spatial/kinematic problems, Sloman's observation that a change in the represented configuration would require many more changes in a Fregean representation than in an analogical representation is of particular importance. This is a severe drawback of Fregean representations for spatial/kinematic problems. Regarding *spatial* (and other) problems, Sloman (1971), Forbus (1981), Forbus (1983), Hinton (1979), Brooks (1981), McDermott and Davis (1984), Larkin and Simon (1987), and Selman (1988) also emphasize the computational complexity savings in certain examples resulting from vivid (and non-vivid) analog processing and/or point out the inefficiency of Fregean representations.

Our approach employs both analogical representations, and analogical processes. Such an approach, then, is *vivid* and *pictorial* in the strongest sense from Selman's survey (

Selman (1988)) on reasoning in Artificial Intelligence with analogs.

Funt (1980) (see also Rumelhart and Norman (1974)), mentioned above in Section 4, designed a diagrammatic representation scheme for qualitative physics (including "mental" rotation). The scheme admitted of exploiting the natural parallelism inherent in moving a multi-particle object around and was based on a 2-dimensional, foveated retina-like spatial framework. (He actually constructed only a sequential simulation, which sufficed for his purposes.) Funt's framework was designed to be especially good for rotation of an object, but its corresponding limitation was that it had but one center of rotation, the center of the framework, where there was maximal resolution. It was not possible, then, to simulate a non-trivial gear train linear in real-time, but, as is pointed out in Brachman and Levesque (1985), the importance of Funt's project was, among other things, in showing that Sloman (1985) need not have partially recanted his position in Sloman (1971) that reasoning with analogs is important to Artificial Intelligence. Our lattice computers are designed to have highly desirable translational invariance properties, and to admit of pretty good rotation about any point. Foveated retinas nicely handle many biological engineering constraints for seeing, but our design doesn't need to be bound by these constraints.

As is pointed out many places, e.g, Minsky (1986), a curious discovery of early Artificial Intelligence is that it was fairly easy to simulate esoteric, expert problem solving in such disciplines as calculus, but very difficult to simulate the perfectly commonplace cognition involved in the *common sense* of a child building block structures or a janitor cleaning a room. *Part* of the trouble with those latter problems is the difficult problem of vision, but, and this is important for our project, it also seems to be very difficult to construct the knowledge bases required. If one *abstracts* out the problem of vision from most (or all) examples of common sense tasks hard for Artificial Intelligence and easy for perfectly ordinary people, one is mostly left with *spatial* reasoning/cognition/puzzle-solving/performance tasks! Here is what we think is going on. *Totally* verbal/propositional representation schemes for general spatial tasks are inherently too complex to build and too inefficient to use! It is much easier to build a spatial framework, and move analog objects around in it to see what will happen. Such a system has the perfect infrastructure for spatial *performance* (or procedural) knowledge. Most of the knowledge of space is, then, stored in the spatial framework! We suspect that that is just a vastly more *succinct* representation than any completely verbal/propositional scheme; furthermore, totally verbal/propositional schemes are, many times, too computationally complex to access and update, but, for example, lattice computers naturally and efficiently exploit parallelism and do *not* have an update propagation problem. We further conjecture that these efficiency issues together with the strong biological need for complex animals to handle spatial problems efficiently forced the human brain to evolve so that, for most people, there is a special region of the brain providing a spatial framework.

While Genesereth and Nilsson devoted Genesereth and Nilsson (1987) to the study of declarative representations in Artificial Intelligence, they clearly recognize the importance of both declarative and procedural/performance representations. They do claim, though, that procedural knowledge is not as flexible as declarative, that procedurally represented knowledge is harder to change. However, they weren't thinking of examples of spatial tasks employing the infrastructure (spatial framework) of lattice computers, but, on their side, see the next paragraph just below. Lattice computers are general purpose for spatial/kinematic knowledge, hence flexible in that respect. The update propagation problem is severe for declarative representations, but clearly our knowledge bases update quite automatically regarding the positions of objects in space and the like.

We expect that our lattice computers will be very good vehicles for modeling the *learning* of one performance task similar to another one already programmed in. See the discussions below on template pattern matching. We don't know yet if performance learning in our lattice computers will itself suffer an update propagation problem, and this is probably closer to Genesereth and Nilsson's concern.

Genesereth and Nilsson also note that declarative knowledge can be accessed introspectively. It turns out, though, that any computation task in any acceptable programming formalism (see Rogers (1958), Rogers (1967), and Machtey and Young (1978))[2] can be handled by an introspective program which explicitly constructs a perfect, quiescent copy of itself to inspect as data. As the first author has pointed out many times (e.g., Case (1993), Case (1986), Royer (1987), Royer and Case (1986), Royer and Case (1991), and Case and Smith (1983)), various recursion theorems from recursive function theory essentially guarantee this. Our lattice computers essentially constitute such an acceptable programming formalism. In fact, in most instances where the "non-empty" portion of the lattice computer resides in n-dimensions and the lattice computer is at least two processors thick in the $(n+1)$-st-dimension, a quiescent copy of the active portion can be projected into the extra dimension in one step and then accessed as data for introspection by the active portion. Hence, introspective programs are not limited to programming formalisms featuring only declarative representations.

In a private conversation, Jerry Feldman, a connectionism advocate, suggested that one of his reasons for being such is that he thinks that performance knowledge is crucial to human cognition and that connectionist networks are good for modeling it. Of course, he, as well as other connectionists, are also interested in using connectionist networks to model declarative knowledge. In a sense, our lattice computers might be construed as a special case of a connectionist network, but we believe that does not give the most useful conceptual picture of what our lattice computers are about.

Lattice computers might be helpers for natural language processing: one could use lattice computers to dynamically store and modify a diagrammatic, "concrete" model of (some of) what's being discussed as the discourse comes in. The diagrams could include time lines. Objects, when first introduced with not yet totally definite features and spatial and temporal placement, could be given *generic*, simple features and spatial and temporal placement consistent with the discourse up to their introduction; then, later, as more information comes in, the analogical representation could be updated by changing/adding features, and analogically moving the objects about in time and space. Objects given generic features, temporal and/or spatial placement because the specific information is not available could be tagged as tentative with respect to this information until (if ever) the information becomes specific. Ambiguities could be handled with multiple analogical models. Since half of a metaphor is relatively concrete, building a (generic) analogous model of that half would be useful in metaphor comprehension. For example, Lakoff and Johnson (1980) introduce the *conduit metaphor* to explain how we understand the abstraction of *communication*: communication can be viewed (metaphorically) as the process of putting an object in a container, moving that container along a conduit, and then taking out the object at the other end of the conduit when the container arrives. So, for example, to *help* understand sentences about communication, an analog of a physical model of Lakoff and Johnson's conduit with containers could be constructed and manipulated in our lattice computers.

In a private conversation the first author had a few years ago with Michael Gazzaniga (a former Sperry student and, at that time, on the faculty at Stony Brook), it developed that he favored the not uncommon point of view (Marvin Minsky's view too, we believe) that the right-hemisphere is just a retarded left-hemisphere. When the first author pressed him on the point of which hemisphere in split-brain patients (his specialty; Sperry got the Nobel Prize for his ground-breaking work with these patients) is superior for such *spatial* tasks as Koh Block Designs, he admitted that the right-hemisphere was. In "cognitive science" there is an overwhelmingly large preponderance of two kinds of people: those who recognize the existence of non-verbal cognition, but who are working on verbal/propositional cognition, and those who don't seem to recognize the existence of non-verbal cognition. We really can't do anything about this state of affairs except to continue our work and be grateful for the existence of those who work on imagery and such like.

Rosch (see Mervis and Rosch (1981) and Rosch and Lloyd (1978)) takes the point of view that those *categories* in which humans naturally think (except, perhaps, in mathematics) are each based, not on a careful verbal description, but on one (or a few) prototypes with membership in the category, then, determined by "closeness" to prototype. For example, for the category of *bird* most people seem to have a wren or robin as prototype bird. Here's some evidence. In reaction time experiments in which people are shown a picture of an animal and asked to respond as quickly as possible to the question, "Is this animal a bird?", their response times are fast on wrens and robins, a little slower on blue jays and cardinals, and very slow on penguins, ostriches, and kiwis. Also they have fast response for elephant and slow response for bat. It is conceivable, but not yet certain, then, that there is an imagistic component to common human conceptualization by categories. The measure of "closeness" may be visual. Perhaps our lattice computers can be a vehicle to model aspects of this usefully. Our lattice computers could be used for template pattern matching. This may, then, eventually be relevant to category testing by approximate matching to prototype. It may also be relevant to prototype formation. Here's how that might work. One overlays (appropriately dimensioned) patterns, each representing an example; prototypes are examples that "best fit" the overlaid pattern of all known examples (see the discussion of Galton's photography experiments below). This sort of thing may be of aid in vision too. Furthermore, regarding vision, it may be helpful, for example, to pattern match a sequence of *projections* from various vantage points of stored 3-dimensional (prototypical?) scenes with (possibly preprocessed) retinal-like images from corresponding vantage points. Of course, there is the following problem for template pattern matching: the patterns must be overlaid just right (perhaps with a topological distortion); expensive trial and error may be required[3] to find a nicely matching overlaying! In some cases, though, there may be a clear way to find points in each pattern to serve as the anchors for overlaying to check matching. For human face patterns, Sir Francis Galton (see, for example, Newman (1961)), in his experiments with superimposed photographs, found that it sufficed, for front views, just to anchor on eyes and mouth. If those lined up, the superpositions of faces looked like faces. His superposition of faces of members of families picked out family resemblance. The superposition of faces of (unrelated) women were strikingly beautiful. He did not have the advantage of being able to do matching for dimension > 2 patterns.

Regarding loose or approximate template matching and the learning or *partial* reproduction of performance (e.g., tennis strokes) we are reminded of Bartlett (1932), Pages 201-213.

Perhaps automatic theorem proving (for robotics and math) could be considerably enhanced by connecting up a logic engine to one of our lattice computers and using the lattice computer to do relevant diagram manipulation to cut down on the combinatorial explosion of pure trial and error on the logic engine side (as in Gelernter (1963)). Even mathematicians who work in a specialty neither inherently pictorial nor about (idealized) pictures, say, a specialty quite unlike geometry, nonetheless, work mentally and/or on paper with (possibly moving) diagrams. These diagrams *metaphorically* (metaphor again!) represent abstractions of the specialty. This is very much the case, for example, for recursive function theory. We suspect it is mainly this spatial manipulative cognitive component that enables human mathematicians to considerably surpass *current* automatic theorem proving systems in sophisticated mathematical theorem proving. It seems likely that, theoretically, the speed-up in finding proofs is not significant as measured by $\mathbf{O}(\ )$ (see Cormen, Leiserson and Rivest (1990)), but does involve some *significantly* large constants, significant relative to, say, human lifetimes.

5.4 Cellular Automata and Physics

We expect that microscopically accurate (discrete, analogical) simulations of the motions of multiple, perhaps totally interacting, objects can be done on our lattice computers, but, some preliminary analysis of simple cases suggests these simulations might be slow (for example, if the forces between them act orders of magnitude more quickly than the bodies can respond), although linear in real time. They shouldn't be much slower, though, for many objects, than for a few. It may also be that simulations of some physical phenomena on our lattice computers may be piecewise (approximately) linear in real-time if simulations have to be stopped once in a while to rescale magnitudes. Case, Rajan and Shende (1990b) cites a feasibility result we obtained suggesting that for modeling closed physical systems with superimposable forces, we should be able to do our lattice computer simulations without swamping the bandwidth of the nearest-neighbor connections, a problem with many numerical analysis simulations.

Real space may not be euclidean; it may, for example, be twisted up a bit. Spaces people have bothered to think up can generally be embedded in n-dimensional euclidean space, for suitably large n. (For example, the surface of a Klein bottle can be embedded in 4-dimensional euclidean space with no self-intersecting surfaces.) Hence, one of our appropriately dimensioned lattice computers would suffice to represent discretized versions of them.

Heisenberg (1958), Pages 164-165 argues that there must exist a universal constant in nature—the smallest length. This might mean that the universe, including space, is discrete. Some of the workers in the cellular automata approach to physics (see, for example, Feynman (1982), Minsky (1982), Fredkin and Toffoli (1982), Toffoli (1984), Toffoli and Margolus (1987), Margolus (1984), Salem and Wolfram (1986), Thinking Machines (1986), Toffoli (1977b), Toffoli (1977a), Vichniac (1984), Wolfram (1983), Svozil (1986), Frisch, Hasslacher and Pomeau (1986), and Hasslacher (1987)) seem to take this idea seriously. The theory of Cellular Automata (Neumann (1966), Burks (1970), and Codd (1968)) is a clean, theoretical model for representing discrete space, especially from the point of view of massively parallel processing. A *cellular automaton* is essentially (by definition) a mesh-work of processors (where each processor is thought of as a finite automaton—see Neumann (1966), Burks (1970), and Codd (1968). Several cellular automata based architectures have been proposed and built for problems in discrete physics (Toffoli and Margolus (1987)), graphics (Poulton, Fuchs and others (1985)), and medical image processing (Preston and Duff (1984), Chaps. 10 & 11). Most of these architectures are 2-dimensional square mesh-works. The philosophy of computing in the cellular automata physics community seems to require that each individual cellular machine (we would say processor) must be extremely simple. To some extent this may be a carry over from the theoretical cellular automata theorists; to some extent this may come from the idea that each point in physical space must be quite simple. *We* are *not* bound by this seeming constraint. It should be admitted that the cellular automata approach does provide processors that are easy to build. In *our* approach, we decided to trade processor complexity (to a feasible extent) for increased flexibility in programming motion.

The Connection Machine, of the Thinking Machines Corp., a massively parallel computer with simple individual processors, employs a hard-wired interconnection scheme quite different from ours. Also different is the philosophy behind the interconnection scheme (and other design features)—they were interested in designing a *general-purpose* parallel architecture. So, they were necessarily sensitive to the issue of their architecture having a small *diameter*, an issue prevalent among those involved in the design of general-purpose parallel architectures (see, for example, Miller and Stout (1989)). For example, one of the operations that is basic in most people's approach to parallel computing is that of *sorting* data in processors which are widely scattered across the whole architecture. For sorting, the path length in the architecture between any two pieces of data becomes of paramount importance because this length places a lower bound on the amount of time required to compare the data that is scattered in the architecture. Keeping the maximum such path length, i.e., the *diameter* of the architecture, as low as possible is clearly, *then*, an important criterion in designing general-purpose parallel architectures. Our lattice computers, on the other hand, should have a *large* diameter to simulate large chucks of space, and, fortunately, we essentially have no need to use operations which dictate low diameter, e.g., operations such as sorting data scattered across a mesh-works. For a given, large number of processors, we also happen to avoid the physical wiring problems associated with tightly-coupled architectures offering a low diameter (see Vitányi (1988)).

While our lattice computer approach bears a resemblance to that of cellular automata approaches, our philosophy of algorithmic locality is also not subject to the same apparent computational limitations. For example, (as noted in Section 4 above) although 2-dimensional *analogical* fluid flow simulations (based on the lattice $\mathbf{A}_2$) can be carried out quite successfully with the cellular automata approach (see Salem and Wolfram (1986) and Thinking Machines (1986)), there are problems (for example using the lattice $\mathbf{A}_3$) in the 3-dimensional case (see Frisch, Hasslacher and Pomeau (1986) and Hasslacher (1987)). We believe our algorithm in Shende (1991) and Case, Rajan and Shende (1991b), lets call it TRANSLATE, to simulate constant speed, straight line particle motion may serve as another basis for circumventing these problems. Hasslacher (1987) notes that projecting from a fourth dimension (with two velocities) helps for *some* cases of fluid flow in the cellular automata approach. We expect our approach will be more general purpose for tackling a *variety* of cases computationally. Essentially TRANSLATE enables us to send particles in *any lattice-representable (virtual) direction* at (approximately) constant speed. In a lattice computer based on $\mathbf{A}_3$, while there are (up to) 12 directions given by the local, nearest-neighbor connections, many more *virtual* directions are representable. The limitations of the cellular automata approach pointed out in Frisch, Hasslacher and Pomeau (1986) and Hasslacher (1987) are fundamentally tied to the geometric shape and symmetry of the *local* connectedness (e.g., the 12-connectedness of $\mathbf{A}_3$). With TRANSLATE we can, in effect, add more virtual directions to try to circumvent these limitations in new ways and add generality. Much remains to be tested. Our attempts would likely be aided by feasible increases in the resource of communication bandwidth between nearest-neighbors to handle extra directions.

6 Notes

1. There is some evidence that some retinotopic maps on the brain are complex logarithm transformations of the retinal image (see Baron (1987)) so that magnification on the retina corresponds to translation in one direction on the representation and rotation on the retina corresponds to translation in a perpendicular direction on the representation. We also recall reading that an experiment on monkeys showed that rotation of an image in the plane of their retinas actually corresponded to a rotation in at least one of the many retinotopic maps on their brains.

2. Case showed the *acceptable* systems are *characterized* as those in which every control structure can be constructed (see Riccardi (1980) and Royer (1987)). Royer and later Marcoux examined complexity analogs of this characterization (see Riccardi (1980), Riccardi (1981), Royer (1987), and Marcoux (1989)). All "naturally" occurring programming systems are essentially acceptable, but non-acceptable systems can be mathematically constructed (see Rogers (1958)).

3. This expense may be reflected in the apparent high computational complexity of the graph isomorphism problem (see Garey and Johnson (1979)).

4. Both baboons and humans show an apparent *left* brain hemisphere advantage for distinguishing between rotated and reversed planar alphabet letters (see Bower (1993) for a report on experiments with baboons by Hopkins, Vauclari and Fagot). Possibly this is a case where the figures are simple enough that a spatial framework is not as useful as a local analysis. In a February 1993 private conversation with the first author, Hopkins pointed to some very recent work by Fagot, again on baboons, but featuring tasks involving large alphabet letters made out of smaller (same or different) alphabet letters. This results point to baboons employing global processing with their right brain hemispheres and local in their left!

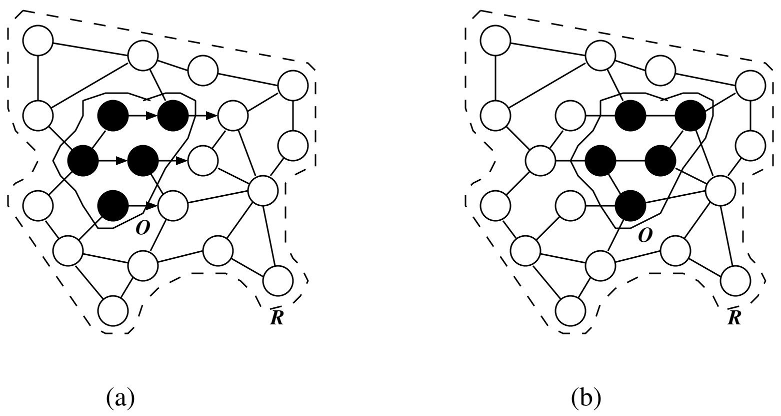

Figure 1. Analogical Representation of the Motion of $\boldsymbol{O}$ in $\boldsymbol{R}$

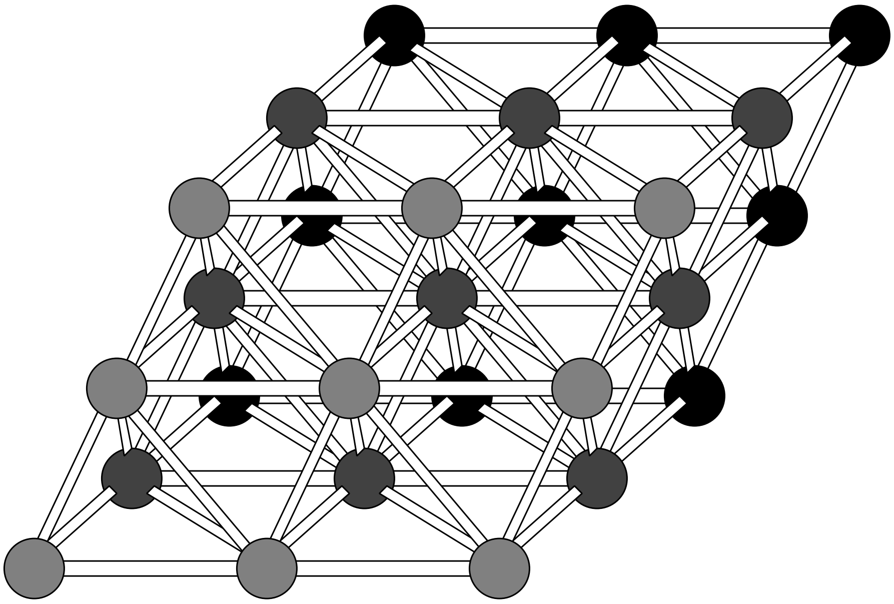

Figure 2. A Small Lattice Computer Based on $\mathbf{A}_3$

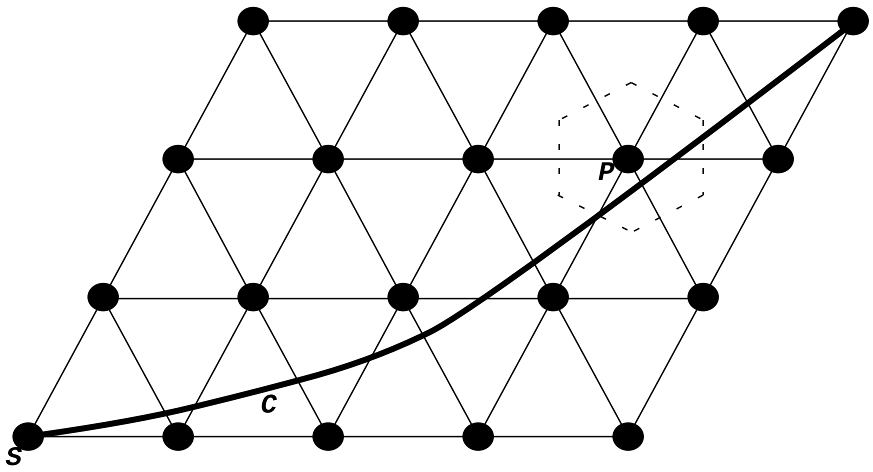

Figure 3. A Curve $\boldsymbol{C}$ and a Voronoi Cell in a Lattice Computer Based on $\mathbf{A}_2$

References

Anthony, R. J. (1938). Essai de recherche d'une expression anatomique approximative du degré d'organisation cérébrale autre que le poids de l'encephale comparée au poids du corps. *Bulletin de la Societé N' Anthropologie de Paris*, *9*, 1–67.

Aurenhammer, F. (1990). Voronoi diagrams—a survey of a fundamental geometric data structure. Technical Report B-90-9, Institute for Computer Science, Dept. of Mathematics, Freie Universität Berlin.

Baron, R. (1985a). A model of mental imagery. *Int. J. Man–Machine Studies*, *23*, 313–334.

Baron, R. (1985b). Visual memories and mental images. *Int. J. Man–Machine Studies*, *23*, 275–311.

Baron, R. (1987). *The Cerebral Computer*. Lawrence Erlbaum, Associates, Hillsdale, NJ.

Bartlett, F. C. (1932). *Remembering — A Study in Experimental and Social Psychology*. Cambridge University Press, Cambridge. Reprinted, 1977.

Blinkov, S. M. & Glezer, I. I. (1968). *The Human Brain in Figures and Tables*. Plenum Press, NY.

Block, N. (Ed.). (1981). *Imagery*. MIT Press, Cambridge, MA.

Bower, B. (1993). Baboons offer glimpses of left-brain brawn. *Science News*, 54.

Brachman, R. & Levesque, H. J. (Eds.). (1985). *Readings in Knowledge Representation*. Morgan Kaufmann Publishers, Inc.

Brooks, R. (1981). Symbolic reasoning among 3–d models and 2–d images. *Artificial Intelligence*, *17*, 285–348.

Burks, A. W. (Ed.). (1970). *Essays on Cellular Automata*. Univ. of Illinois Press.

Case, J. (1986). Learning machines. In W. Demopoulos & A. Marras (Eds.), *Language Learning and Concept Acquisition: Foundational Issues*. Ablex Publishing Company, Norwood, NJ.

Case, J. (1987). Turing machine. In S. Shapiro (Ed.), *Encyclopedia of Artificial Intelligence*. John Wiley and Sons, New York, NY.

Case, J. (1993). Infinitary self-reference in learning theory. *Journal of Experimental and Theoretical Artificial Intelligence*. To appear in the special issue from AII 92.

Case, J., Chitoor, S., Rajan, D. S., & Shende, A. M. (1993). Writeup in preparation.

Case, J., Rajan, D. S., & Shende, A. M. (1990a). Optimally representing euclidean space discretely for analogically simulating physical phenomena. In Nori, K. V. (Ed.), *Foundations of Software Technology & Theoretical Computer Science*, volume 472 of *Lecture Notes in Computer Science*, (pp. 190–203). Springer Verlag, Berlin. See Case, Rajan and Shende (1991a).

Case, J., Rajan, D. S., & Shende, A. M. (1990b). Parallel processor computer. U.S. Patent Application submitted.

Case, J., Rajan, D. S., & Shende, A. M. (1991a). Lattice computers for optimally representing euclidean space. Technical Report 91-15, University of Delaware. Expands on and corrects slightly Case, Rajan and Shende (1990a).

Case, J., Rajan, D. S., & Shende, A. M. (1991b). Simulating uniform motion in lattice computers I: Constant speed particle translation. Technical Report 91-17, University of Delaware.

Case, J., Rajan, D. S., & Shende, A. M. (1991c). Spherical wave front generation in lattice computers. Technical Report 91-16, University of Delaware.

Case, J. & Smith, C. (1983). Comparison of identification criteria for machine inductive inference. *Theor. Comp. Sci.*, *25*, 193–220.

Chandrupatla, T. & Belegundu, A. (1991). *Introduction to Finite Elements in Engineering*. Prentice-Hall, NJ.

Codd, E. F. (1968). *Cellular Automata*. Academic Press.

Conway, J. H. & N. J. A. Sloane (1993). *Sphere Packings, Lattices and Groups* (Second ed.). Springer Verlag.

Cook, N. D. (1986). *The Brain Code: Mechanisms of Information Transfer and the Role of the Corpus Callosum.* Methuen, London.

Cormen, T., Leiserson, C., & Rivest, R. (1990). *Introduction to Algorithms.* MIT Press/McGraw Hill.

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics*, *21*(6/7).

Finke, R. A. (1980). Levels of equivalence in imagery and perception. *Psychological Review*, *87*, 113–139.

Finke, R. A. & Schmidt, M. J. (1977). Orientation–specific color after–effects following imagination. *Journal of Experimental Psychology: Human Perception and Performance*, *3*, 599–606.

Flynn, M. & Kosaraju, R. (1976). *Kybernetes*, *5*, 159–163.

Forbus, K. (1981). A study of qualitative and geometric knowledge in reasoning about motion. Technical Report AI.TR.615, MIT, AI–Lab.

Forbus, K. (1983). Qualitative reasoning about space and motion. In D. Gentner & A. Stevens (Eds.), *Mental Models.* Lawrence Erlbaum Associates, Hillsdale, N.J.

Fredkin, E. & Toffoli, T. (1982). Conservative logic. *International Journal of Theoretical Physics*, *21*(3/4).

Frisch, U., Hasslacher, B., & Pomeau, Y. (1986). Lattice-gas automata for the Navier Stokes equation. *Physical Review Letters*, *56*(14), 1505–1508.

Funt, B. V. (1980). Problem–solving with diagrammatic representations. *Artificial Intelligence*, *13*, 201–230. Reprinted in Brachman and Levesque (1985).

Gardner, H. (1975). *The Shattered Mind: The Person after Brain Damage.* Knopf, New York.

Gardner, H. (1982). *Art, Mind, and Brain: A Cognitive Approach to Creativity.* Basic Books, New York.

Garey, M. & Johnson, D. (1979). *Computers and Intractability.* Freeman.

Gelernter, H. (1963). Realization of a geometry theorem–proving machine. In E. Feigenbaum & J. Feldman (Eds.), *Computers and Thought.* McGraw–Hill, NY.

Genesereth, M. & Nilsson, N. (1987). *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann, Los Altos, CA.

Gruber, P. M. & Lekkerkerker, G. (1987). *Geometry of Numbers.* North-Holland Mathematical Library.

Hasslacher, B. (1987). Discrete fluids. *Los Alamos Science*, (15), 175–217. Special Issue.

Heisenberg, W. (1958). *Physics and Philosophy.* Harper & Brothers Publishers, New York.

Herstein, I. N. (1964). *Topics in Algebra.* Blaisdell Publishing Co.

Hinton, G. (1979). Some demonstrations of the effects of structural descriptions in mental imagery. *Cognitive Science*, *3*, 231–250.

Kinsbourne, M. (1982). Hemispheric specialization and the growth of human understanding. *American Psychologist*, *37*(4), 411–420.

Kochevar, P. (1991). A simple light simulation algorithm for massively parallel machines. *Journal of Parallel and Distributed Computing*, *13*(2), 193–201.

Kosslyn, S. M. (1980). *Image and Mind.* Harvard Univ. Press, Cambridge, Massachusetts.

Kosslyn, S. M. (1983). *Ghosts in the Mind's Machine: Creating and Using Images in the Brain.* Harvard Univ. Press, Cambridge, Massachusetts.

Kung, H. T. & Leiserson, C. E. (1978). Systolic arrays (for VLSI). In *Proceedings of the Symposium on Sparse Matrix Computing and Their Applications*, (pp. 245–282).

Lakoff, G. & Johnson, M. (1980). *Metaphors We Live By.* University of Chicago Press.

Larkin, J. & Simon, H. (1987). Why a picture is (sometimes) worth ten thousand words. *Cognitive Science.*

Machtey, M. & Young, P. (1978). *An Introduction to the General Theory of Algorithms.* North Holland, New York.

Marcoux, Y. (1989). Composition is almost as good as s-1-1. In *Proceedings, Structure in Complexity Theory–Fourth Annual Conference.* IEEE Computer Society Press.

Margolus, N. (1984). Physics–like models of computation. *Physica 10D*, 81–95.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman, San Francisco.

McDermott, D. & Davis, E. (1984). Planning routes through uncertain territory. *Artificial Intelligence*, *22*, 107–156.

Mervis, C. B. & Rosch, E. (1981). Categorization of natural objects. In M. R. Rosenzweig & L. W. Porter (Eds.), *Annual Review of Psychology*, volume 32.

Miller, R. & Stout, Q. F. (1989). Mesh computer algorithms for computational geometry. *IEEE Transactions on Computers*, *C-38*(3), 321–340.

Minsky, M. (1982). Cellular vacuum. *International Journal of Theoretical Physics*, *21*(6/7).

Minsky, M. (1986). *The Society of Mind.* Simon and Schuster, NY.

Minsky, M. & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry.* MIT Press, Cambridge, MA.

Neumann, J. V. (1966). *Theory of Self–Reproducing Automata.* Univ. Illinois Press. Edited and completed by A. W. Burks.

Newman, J. R. (1961). *Science and Sensibility.* Simon and Schuster, NY.

Nivan, W. D. (Ed.). (1961). *The Scientific Papers of James Clerk Maxwell.* Dover Publications, Inc.

O'Neil, J. (1981). *Prodigal Genius: The Life of Nikola Tesla.* Angriff Press, Hollywood, CA.

Ornstein, R. (1972). *The Psychology of Consciousness.* Freeman, San Francisco, California.

Poulton, J., Fuchs, H., et al. (1985). PIXEL–PLANES: Building a VLSI–based graphic system. In *Chapel Hill Conference on VLSI.*

Preston, K. & Duff, M. J. B. (1984). *Modern Cellular Automata: Theory and Applications.* Plenum Publishers.

Rajan, D. S. & Shende, A. M. (1991). Efficiently generated lattices and root lattices. Technical Report 92-02, University of Delaware. Journal article under review.

Riccardi, G. (1980). *The Independence of Control Structures in Abstract Programming Systems.* PhD thesis, SUNY Buffalo.

Riccardi, G. (1981). The independence of control structures in abstract programming systems. *Journal of Computer and System Sciences*, *22*, 107–143.

Rogers, H. (1958). Gödel numberings of partial recursive functions. *Journal of Symbolic Logic*, *23*, 331–341.

Rogers, H. (1967). *Theory of Recursive Functions and Effective Computability.* McGraw Hill, New York. Reprinted, MIT Press, 1987.

Rosch, E. & Lloyd, B. (Eds.). (1978). *Cognition and Categorization.* Erlbaum Associates, Hillsdale, NJ.

Royer, J. (1987). *A Connotational Theory of Program Structure.* Lecture Notes in Computer Science 273. Springer Verlag.

Royer, J. & Case, J. (1986). Progressions of relatively succinct programs in subrecursive hierarchies. Technical Report 86–007, Univ. of Chicago, Computer Science Dept.

Royer, J. & Case, J. (in revision, 1991). *Intensional Subrecursion and Complexity Theory.* Research Notes in Theoretical Science. Pitman Press.

Rumelhart, D. & Norman, D. (1974). Representation in memory. In R. Atkinson, R. Herrnstein, G. Lindzey, & D. Luce (Eds.), *Stevens' Handbook of Experimental Psychology* (Second ed.)., volume 2: Learning and Cognition. John Wiley and Sons, NY.

Salem, J. B. & Wolfram, S. (1986). Thermodynamics and hydrodynamics with cellular automata. In S. Wolfram (Ed.), *Theory and Applications of Cellular Automata.* World Scientific.

Schilpp, P. A. (Ed.). (1949). *Albert Einstein: Philosopher–Scientist.* Open Court, La Salle, IL.

Selman, B. (1988). Analogues. Technical report, University of Toronto. Technical Note CSRI–47.

Shende, A. M. (1991). *Digital Analog Simulation of Uniform Motion in Representations of Physical n-Space by Lattice-Work MIMD Computer Architectures.* PhD thesis, SUNY at Buffalo, Buffalo, N. Y. TR 91-14, Department of Computer Science.

Shepard, R. N. (1978). The mental image. *American Psychologist, 33*, 123–137.

Shepard, R. N. & Chipman, S. (1970). Second–order isomorphism of internal representations: shapes of states. *Cognitive Psychology, 1*, 1–17.

Shepard, R. N. & Cooper, L. A. (1982). *Mental Images and Their Transformations.* MIT Press, Cambridge, Massachusetts.

Shepard, R. N. & Feng, C. (1972). A chronometric study of mental paperfolding. *Cognitive Psychology, 3*, 228–243.

Shepard, R. N. & Metzler, J. (1971). Mental rotation of three–dimensional objects. *Science, 171*(4), 701–703.

Siegel, H. J. (1985). *Interconnection Networks for Large-Scale Parallel Processing.* Lexington Books.

Sloman, A. (1971). Interactions between philosophy and A.I.—the role of intuition and non–logical reasoning in intelligence. *Artificial Intelligence, 2*, 209–224.

Sloman, A. (1985). Afterthoughts on analogical representations. In R. Brachman & H. J. Levesque (Eds.), *Readings in Knowledge Representation.* Morgan Kaufmann Publishers, Inc.

Sperry, R. W. (1974). Lateral specialization in the surgically separated hemispheres. In B. Milner (Ed.), *Hemispheric Specialization and Interaction.* MIT Press.

Springer, S. & Deutsch, G. (1985). *Left Brain, Right Brain.* W.H. Freeman, New York.

Svozil, K. (1986). Are quantum fields cellular automata? *Physics Letters A, 119*(4).

Thinking Machines (1986). Introduction to data level parallelism. Technical Report 86.14, Thinking Machines.

Toffoli, T. (1977a). Cellular automata machines. Technical Report 208, Comp. Comm. Sci. Dept., University of Michigan.

Toffoli, T. (1977b). Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences, 15*, 213–231.

Toffoli, T. (1984). CAM: A high–performance cellular–automaton machine. *Physica 10D*, 195–204.

Toffoli, T. & Margolus, N. (1987). *Cellular Automata Machines.* MIT Press.

Vichniac, G. (1984). Simulating physics with cellular automata. *Physica 10D*, 96–116.

Vitányi, P. M. B. (1988). Locality, communication and interconnect length in multicomputers. *SIAM Journal on Computing*, *17*(4), 659–672.

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics*, *55*(3), 601–644.