On Distinguishing Epistemic from Pragmatic Action

David Kirsh and **Paul Maglio**
Department of Cognitive Science
University of California, San Diego

Address correspondence to:
David Kirsh
Department of Cognitive Science
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0515

Running Head: Epistemic and Pragmatic Action

ABSTRACT

We present data and argument to show that in Tetris—a real-time, interactive video-game—certain cognitive and perceptual problems are more quickly, easily, and reliably solved by performing actions in the world than by performing computational actions in the head alone. We have found that some of the translations and rotations made by players of this video-game are best understood as actions that use the world to improve cognition. These actions are not used to implement a plan, or to implement a reaction; they are used to change the world in order to simplify the problem-solving task. Thus, we distinguish pragmatic actions—actions performed to bring one physically closer to a goal—from epistemic actions—actions performed to uncover information that is hidden or hard to compute mentally.

To illustrate the need for epistemic actions, we first develop a standard information-processing model of Tetris-cognition, and show that it cannot explain performance data from human players of the game—even when we relax the assumption of fully sequential processing. Standard models disregard many actions taken by players because they appear unmotivated or superfluous. However, we describe many such actions that are actually taken by players that are far from superfluous, and that play valuable roles in improving human performance. We argue that traditional accounts are limited because they regard action as having a single function: to change the world. By recognizing a second function of action—an epistemic function—we can explain many of the actions that a traditional model cannot. Although, our argument is supported by numerous examples specifically from Tetris, we outline how the new category of epistemic action can be incorporated into theories of action more generally.

ON DISTINGUISHING EPISTEMIC FROM PRAGMATIC ACTION

In this paper we introduce the general idea of an *epistemic action*, and discuss its role in Tetris, a real-time, interactive video-game. Epistemic actions—physical actions that make mental computation easier, faster, or more reliable—are *external* actions that an agent performs to change its own computational state.

The biased belief among students of behavior is that actions create physical states which physically advance one towards goals. Through practice, good design, or by planning, intelligent agents regularly bring about goal-relevant physical states quickly or cheaply. It is understandable, then, that studies of intelligent action typically focus on how an agent chooses physically useful actions. Yet, as we will show, not all actions performed by well-adapted agents are best understood as useful physical steps. At times, an agent ignores a physically advantageous action and chooses instead an action that seems physically disadvantageous. When viewed from a perspective which includes epistemic goals—for instance, simplifying mental computation—such actions once again appear to be a cost-effective allocation of the agent's time and effort.

The notion that external actions are often used to simplify mental computation is commonplace in tasks involving the manipulation of external symbols. In algebra, geometry, and arithmetic, for instance, various intermediate results—which could, in principle, be stored in working memory—are recorded externally to reduce cognitive loads (Hitch, 1978). In musical composition (Lerdahl & Jackendoff, 1983), marine navigation (Hutchins, 1990), and a host of expert activities too numerous to list, performance is demonstrably worse if agents rely on their private memory or on their own computational abilities without the help of external supports. Much current research on representation and human computer interface, accordingly, highlights the need to understand the interdependence of internal and external structures (Norman, 1988).

Less widely appreciated is how valuable external actions can be for simplifying the mental computation that takes place in tasks which are not clearly symbolic—particularly in tasks requiring agents to react quickly. We have found that in a video-game as fast paced and reactive as Tetris, the actions of players are often best understood as serving an epistemic function:

the best way to interpret the actions is not as moves intended to improve board position, but rather as moves that simplify the player's problem-solving task.

More precisely, we use the term *epistemic action* to designate a physical action whose primary function is to improve cognition by:

1. reducing the memory involved in mental computation, i.e., space complexity;
2. reducing the number of steps involved in mental computation, i.e., time complexity;
3. reducing the probability of error of mental computation, i.e., unreliability.

Typical epistemic actions found in everyday activities have a longer time-course than those found in Tetris. These include familiar memory-saving actions such as reminding, e.g., placing a key in a shoe, or tying a string around a finger; time-saving actions, such as preparing the workplace, e.g., partially sorting nuts and bolts before beginning an assembly task in order to reduce later search (a similar form of complexity reduction has been studied under the rubric "amortized complexity", Tarjan, 1985); and information gathering activities such as exploring, e.g., scouting unfamiliar terrain to help decide where to camp for the night.

Let us call actions whose primary function is to bring the agent closer to its physical goal *pragmatic actions*, to distinguish them from epistemic actions. As suggested earlier, existing literature on planning (Tate, Hendler & Drummond, 1990), action theory (Bratman, 1987), and to a lesser extent on decision theory (Chernoff & Moses, 1967), has focused almost exclusively on pragmatic actions. In such studies, actions are defined as transformations in physical or social space. The point of planning is to discover a series of transformations that can serve as a path from initial to goal state. The metric of goodness which planners rely on may be the distance, time, or energy required in getting to the goal, an approximation of these, or some measure of the riskiness of the paths. In each case, a plan is a sequence of pragmatic actions justified with respect to its adequacy along one or another of these physical metrics.

Recently, as theorists have become more interested in reactive systems, and in robotic systems that must intelligently regulate their intake of environmental information, the set of actions an agent may perform has been broadened to include perceptual as well as pragmatic actions (see for example, Simmons et al., 1992). However, these inquiries have tended to focus on the control of gaze (the orientation and resolution of a sensor), or on the control of attention (the selection of elements within an image for future processing, Chapman, 1989), as the means of selecting information. Our concern in this paper is with *control of activity.* We wish to know how an agent can use ordinary actions—not sensor actions—to unearth valuable information that is currently unavailable, hard to detect, or hard to compute.

One significant consequence of recognizing epistemic action as a category of activity is that if we continue to view planning as state-space search, we must redefine the state-space in which planning occurs. That is, instead of interpreting the nodes of a state-space graph to be physical states, we have to interpret them as representing both *physical* and *informational states.* In this way, we can capture the fact that a sequence of actions may, at the same time, return the physical world to its starting state, and significantly alter the player's informational state. To preview a Tetris example, a player who moves a piece to the left of the screen then reverses it back to its original position, performs a series of actions that leave the physical state of the game unchanged. By making those moves, however, the player may learn something or succeed in computing something that is worth more than the time lost by the reversal. In order to capture this idea in a form that allows us to continue using our classical models of planning, we must redefine the search-space so that states arrived at after such actions are not identical to earlier states. We will elaborate on this in the discussion section.

Why Tetris?

We have chosen Tetris as a research domain for three reasons. First, it is a fast, repetitive game requiring split second decisions of a perceptual and cognitive sort. Because time is at a premium in this game, a standard performance model would predict that players develop strategies that minimize the number of moves, creating sequences of pragmatic actions that head directly toward goal states. Thus, if epistemic actions are found in the time-limited context of Tetris, they are likely to be found almost everywhere. Second, every action in this game has the effect of bringing a piece either closer to its final position or farther from its final position, so it is easy to distinguish moves that serve a pragmatic function from those that do not. Third, be-

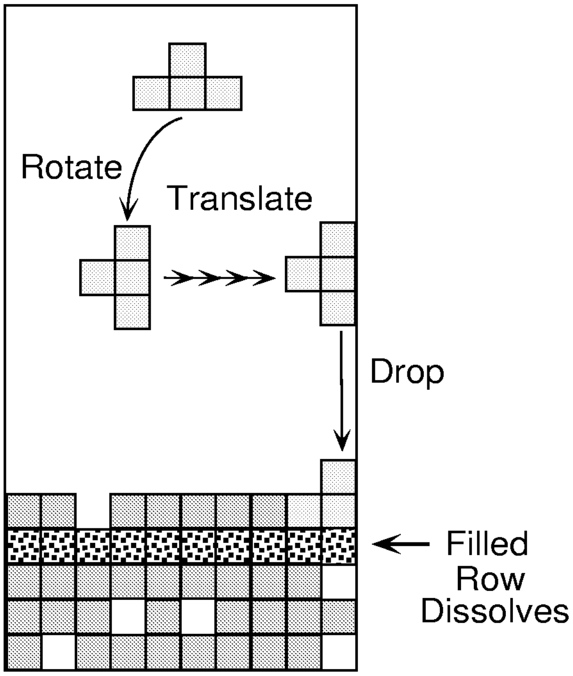
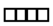
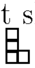


Figure 1. In Tetris, shapes, which we call zoids, fall one a time from the top of the screen, eventually landing on the bottom or on top of shapes that have already landed. As a shape falls, the player can rotate it, translate it to the right or left, or immediately drop it to the bottom. When a row of squares is filled all the way across the screen, it disappears and all rows above it drop down.

cause Tetris is fun to play, it is easy to find advanced subjects willing to play under observation, and easy to find novice subjects willing to practice until they become experts.

Playing Tetris involves maneuvering falling shapes into specific arrangements on the screen. There are seven different shapes, which we call **Tetrazoids**, or simply **zoids**: , , , , , , . These zoids fall one at a time from the top of a screen that is 10 squares wide and 30 squares high (see Figure 2). Each zoid's free-fall continues until it lands on the bottom edge of the screen or on top of a zoid that has already landed. Once a zoid hits its resting place, another zoid begins falling from the top, starting the next Tetris episode. While a zoid is falling, the player can **rotate** it 90° counterclockwise with a single keystroke, or translate it to the **right** or to the **left** one square with a single keystroke. To gain points, the player must find ways of placing zoids so that they fill up rows. When a row fills up with squares all the way across the screen, it disappears and all the rows above it drop down. As more rows are filled, the game speeds up (from an initial free-fall rate of about 200 ms per square to a maximum of about 100 ms per square), and achieving good placements becomes increasingly difficult. As unfilled rows become buried under poorly placed zoids, the squares pile up, creating an uneven contour along the top of the fallen sqaures. The game ends when the screen becomes clogged with these incomplete rows, and new zoids cannot begin descending from the top.

In addition to the rotation and translation actions, the player can **drop** a falling zoid instantly to the bottom, effectively placing it in the position it would eventually come to rest in if no more keys were pressed. Dropping is an optional maneuver, and not all players use it. Dropping is primarily used to speed up the pace of the game, creating shorter episodes without affecting the free-fall rate.

There are only four possible actions a player can take: translate a zoid right, translate left, rotate, and drop. Because the set of possible actions is so small, the game is not very difficult to learn. In fewer than 10 hours, a newcomer can play at an intermediate level. The game is challenging, even for experts, because its pace—the free-fall rate—increases with a player's score, leaving less and less time to make the judgements involved in choosing and executing a placement. This speed-up puts pressure on the motor, perceptual, and reasoning systems, for in order to improve performance, players must master the mapping between keystrokes and effect (motor skills), learn to recognize zoids quickly despite orientation (perceptual skills), and acquire the spatial reasoning skills involved in this type of packing problem.

In studying Tetris-playing, we have gathered three sorts of data:

1. We have implemented a computational laboratory that lets us unobtrusively record the timing of all keystrokes and game situations of subjects playing Tetris.
2. We have collected tachistoscopic tests of subjects performing mental rotation tasks related to Tetris.

3. We have designed and implemented an expert system to play Tetris and have compared human and machine performance along a variety of dimensions.

In what follows, we use these data to argue that standard accounts of practiced activity are misleading simplifications of whatever processes actually underlie performance. For instance, standard accounts of skill acquisition explain enhanced performance as the result of chunking, caching, or compiling (Newell, 1990; Newell & Rosenbloom, 1981; Reason, 1990; Anderson, 1983). Although our data suggest that Tetris-playing is highly automated, we cannot properly understand the nature of this automaticity unless we see how closely action is coupled to cognition. Agents do not simply cache associative rules describing what to do in particular circumstances. If caching were the source of improvement, efficiency would accrue from following roughly the same cognitive strategy used before caching, only doing it faster because the behavioral routines are compiled. If chunking were the source of improvement, efficiency would accrue from eliminating intermediate steps, leading sometimes to more far-reaching strategies, but ones nonetheless similar in basic style. Our observations, however, is that agents learn qualitatively different behavioral tricks. Agents learn how to expose information early, how to prime themselves to recognize zoids faster, and how to perform external checks or verifications to reduce the uncertainty of judgements. Of course, such epistemic procedures may be cached, but they are not pragmatic procedures; they are procedures that direct the agent to exploit its environment to make the most of its limited cognitive resources.

To make this case, we begin by briefly constructing a classical information-processing account of Tetris-cognition and show that it fails to explain, even coarsely, some very basic empirical facts about how people play. We then distinguish more carefully several different epistemic functions of actions in Tetris, showing how these presuppose a tighter coupling of action and cognition. We conclude with a general discussion of why epistemic action is an important idea, and how it might be exploited in the future.

A PROCESS MODEL

RoboTetris is a program we have implemented to help us computationally explore the basic cognitive problems involved in playing Tetris. It is based on a classical information-processing model of expertise that supposes Tetris-cognition proceeds in four major phases:

1. Create an early, bitmap representation of selected features of the current situation.
2. Encode the bitmap representation in a more compact, chunked, symbolic representation.
3. Compute the best place to put the zoid.
4. Compute the trajectory of moves to achieve the goal placement.

Figure 2 graphically depicts this model.

Phase One: Create Bitmap

Light caused by the visual display strikes the retinal cortex and initiates early visual processing. Elaborate parallel neural computation extracts context-dependent features and represents them in a brief sensory memory, often called an iconic buffer (Sperling, 1960; Neisser, 1967). The contents of the iconic buffer are similar to maps, in which important visual features, such as contours, corners, colors, etc., are present but not encoded symbolically. That is, the memory regions that carry information about color and line segments are not *labelled* by symbol structures indicating the color, kind of line segment, or any other attributes present, such as length and width. Rather, such information is extractable but additional processing is required to encode it in an explicit or usable form.[1]

Phase Two: Create Chunked Representation

By attending to sub-areas of iconic memory, task-relevant features are extracted and explicitly encoded in working memory. To make the discussion of RoboTetris concrete, we introduce its symbolic representation which includes features similar to the line-labelling primitives used by Waltz (1975):

[1] For one account of what it means for information to be explicitly encoded, see Kirsh (1990).

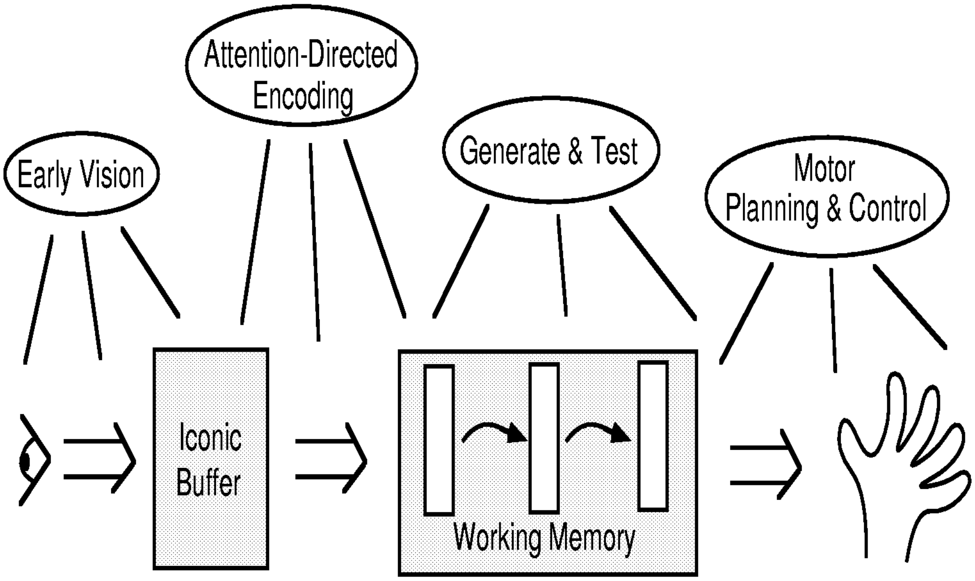


Figure 2. In our classical information-processing model of Tetris-cognition, first a bitmap-like representation floods the iconic buffer, then attention selectively examines this map to encode zoid and contour chunks. These chunks accumulate in working memory, providing the basis for an internal search for the best place to put the zoid. This search can be viewed as a process of generating and evaluating possible placements. Once a placement has been chosen, a motor plan for reaching the target is computed. The plan is then handed off to a motor controller for regulating muscle movement.

concave corners, convex corners, and **T-junctions** (see Figure 3). Such a representation has advantages, but our argument does not rely critically on this choice. Another set of symbolic features might serve just as well, provided that it too can be computed from pop-out features—such as line segments, intersections, and shading (or color)—by selectively directing attention to conjunctions of these (Treisman & Souther, 1985), and that it facilitates the matching process of Phase Three.

As yet, we do not know if skilled players encode symbolic features more quickly in working memory than less skilled players. Such a question is worth asking, but regardless of the answer, we expect that absolute speed of symbolic encoding is a less significant determinant of performance than the size of the chunks encoded. *Chunks* are organized or structured collections of features which regularly recur in play. They can be treated as labels for rapidly retrievable clusters of features which better players use for encoding both zoids and contours (see Figure 4). As in classical theory, we assume that much of expertise consists in refining selective attention to allow ever larger chunks of features to be recognized rapidly

Given the importance of chunking, a key requirement for a useful feature language—one provably satisfied by our line-labelling representation—is that it is expressive enough to uniquely encode every orientation of every zoid, and to allow easy expression of the constraints on matching that hold when determining whether a particular chunk fits snugly into a given fragment of contour (see Figure 5).

Phase Three: Determine Placement

Once zoid and contour are encoded in symbolic features and chunks, they can be compared in working memory to identify the best region of the contour on which to place the zoid. Later in the paper we will mention some alternative ways this matching may unfold. In RoboTetris, the general process is to search for the largest uninterrupted contour segment that the zoid can fit, and to weigh this candidate placement against others on the basis of a set of additional factors, such as how flat the resultant contour will be, how many points will be gained by the placement, and so on. Because both zoids and contours are represented as collections of chunks, finding a good

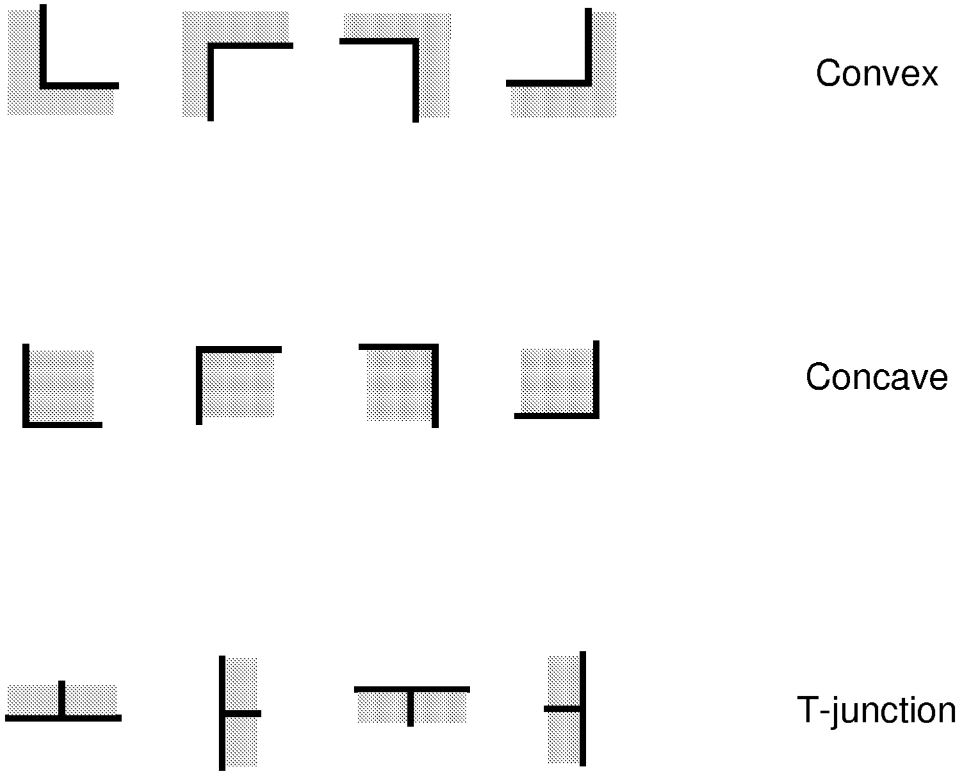


Figure 3. Three general features—concave, convex, T-junction—in each of their orientations create twelve distinct, orientation-sensitive features. These features are extracted by selectively attending to conjuctions of the more primitive features: lines, intersections, and shading.

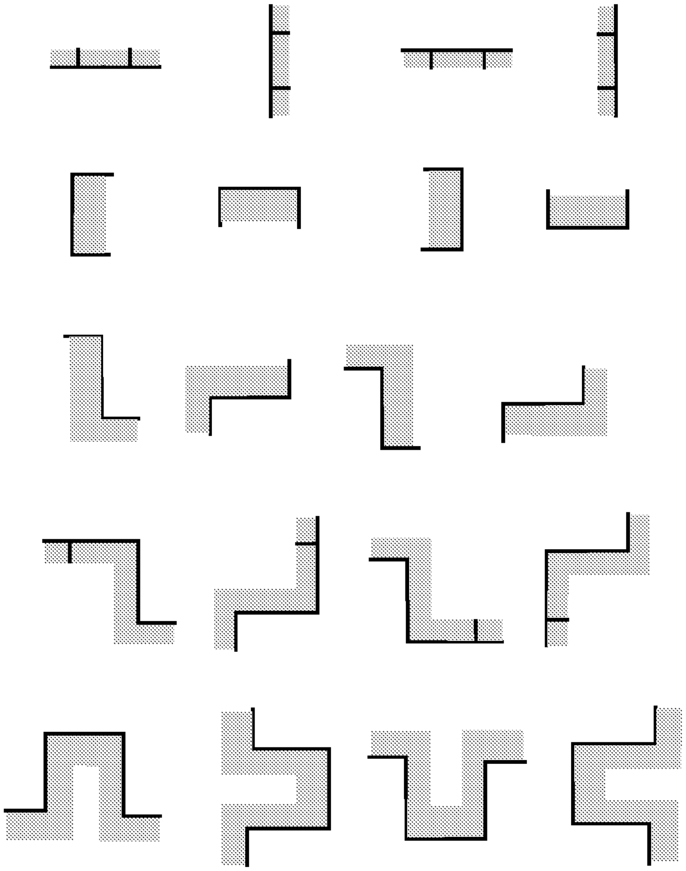


Figure 4. The greater a player's expertise, the more skilled the perception. This is reflected by the size and type of the chunked features which attention-directed processes are able to extract from iconic memory. This figure shows chunks of different sizes and types. Each chunk is a structured collection of primitive features.

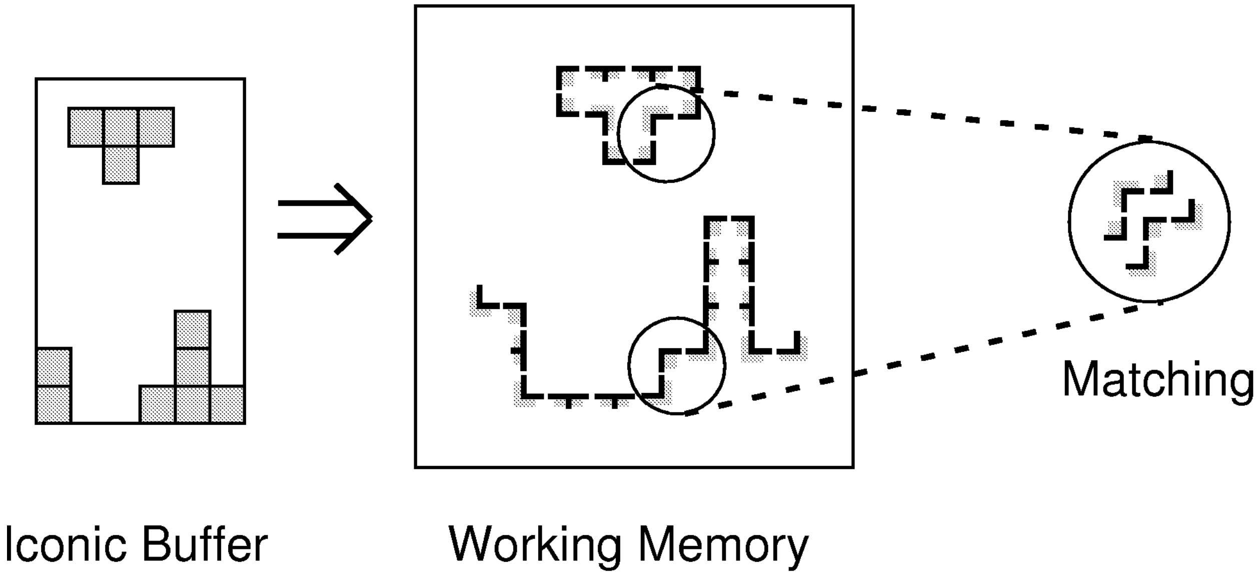


Figure 5. A good representation must make it easy to recognize when zoid and contour fragments match. In this figure, a zoid chunk matches a contour chunk when concave corners match convex corners and straight edges match straight edges. This simple complimentarity is probably computed in the visuo-spatial component of working memory (Baddeley, 1990).

placement involves matching chunks to generate candidate locations. To test the candidates, actual placements are simulated in an internal model of the Tetris situation.

Phase Four: Compute Motor Plan

Once a target placement is determined, it is possible to compute a sequence of actions (or equivalently, keystrokes) that will maneuver the zoid from its current orientation and position to its final orientation and position. The generation of this motor plan occurs in Phase Four. We assume that such a motor plan will be minimal in that it specifies just those rotations and translations necessary to appropriately orient and place the zoid.

After Phase Four, RoboTetris carries out the motor plan by directly affecting the ongoing Tetris game, effectively hitting a sequence of keys to take the planned action.

This completes our brief account of how a classical information-processing theorist might try to explain human performance, and how we have designed RoboTetris on these principles.

How Realistic is this Model?

As we have stated it, the model is fully sequential: Phase Two is completed before Phase Three begins, and Three is completed before Four begins. Because all processing within Phase Four must also be completed before execution begins, the muscle control system cannot receive signals to begin movements until a complete plan has been formulated. Any actions we find occuring before the processing of Phase Four is complete must, in effect, be unplanned; they cannot be under *rational* control and so ought, in principle, to be no better than random actions.

This is patently not what we see in the data. Rotations and translations occur in abundance, almost from the moment a zoid enters the Tetris screen. If players actually wait until they have formulated a plan before they act, the number of rotations should average to half the number of rotations that can be performed on the zoid before an orientation repeats. This follows because each zoid emerges in a random orientation, and on average, any zoid can be expected to be placed in any of its orientations with equal probability. Thus, a shape such as , which has four distinct orientations and can be rotated three times before repeating an orientation, ought to average out to 1.5 rotations. As can be seen in Figure 6, each zoid is rotated more than half its possible rotations. And as Figure 7 shows, rotations sometimes begin extremely early, well before an agent could finish thinking about where to place the zoid.

If we wish to save the model within the classical information-processing framework, one obvious step is to allow Phase Four to *overlap* with Phase Three. Instead of viewing Tetris-cognition as proceeding serially, we can view it as a *cascading* process in which each phase begins its processing before it has been given all the information it will eventually receive. In that case, an agent will regularly move zoids before completing deliberation. The simplest way to capture this notion is to suppose that Phase Three constantly provides Phase Four with its *best estimate* of the final choice. Phase Four then begins computing a path to that spot and the agent initiates a response as soon as Phase Four produces its first step.

In the AI planning literature, the analog of cascade processing is *interleaving* (Ambros-Ingerson & Steel, 1987). Interleaving planners begin execution before they have settled on all the details of a plan. Whereas an orthodox planner executes only after formulating a totally ordered list of subgoals, and hence a complete trajectory of actions, an interleaving planner executes its first step before it has completely ordered subgoals, and before it has built a full contingency table for determining how to act. The net effect is that actions are taken on the best current estimate of the plan.

Interleaving is a valuable strategy for coping with a dynamic, hard to predict world. When the consequences of action cannot be confidently predicted, it is wise to update one's plan as soon as new information arrives. Interleaving planners work just that way; they make sense when it seems inevitable that plans will have to be re-evaluated after each action, and modifications made to adapt the ongoing plan to the new circumstances.

Yet in Tetris, the consequences of an action do not change from moment to moment. The effects of rotating a zoid are wholly determinate. The point of interleaving in Tetris, then, cannot be to allow a player to revise his or her plan on the basis of new information about the state of the world. Rather, the point must be to minimize the danger of having too little time for execution. If a player has a good idea early on as to where to place a zoid,

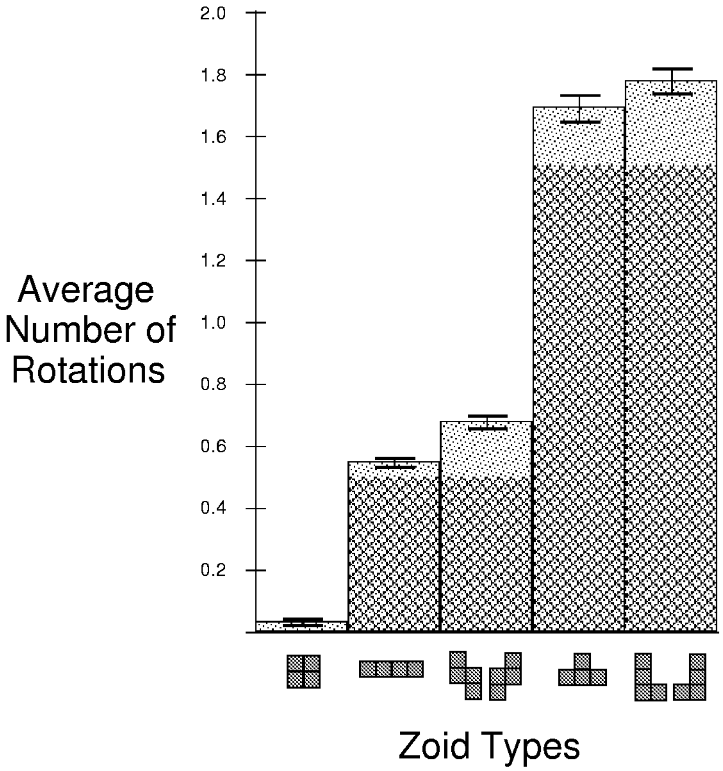


Figure 6. This bar graph shows the average number of rotations for each type of zoid from the moment it emerged to the moment it settled into place. Zoids such as are rotated significantly more than , and both types are rotated more than the expected number of rotations, shown by the crosshatched portions of the bars. Similarly, zoids such as are rotated more than , and both exceed the expected number required for purely pragmatic reasons. The error bars indicate 95% confidence intervals.

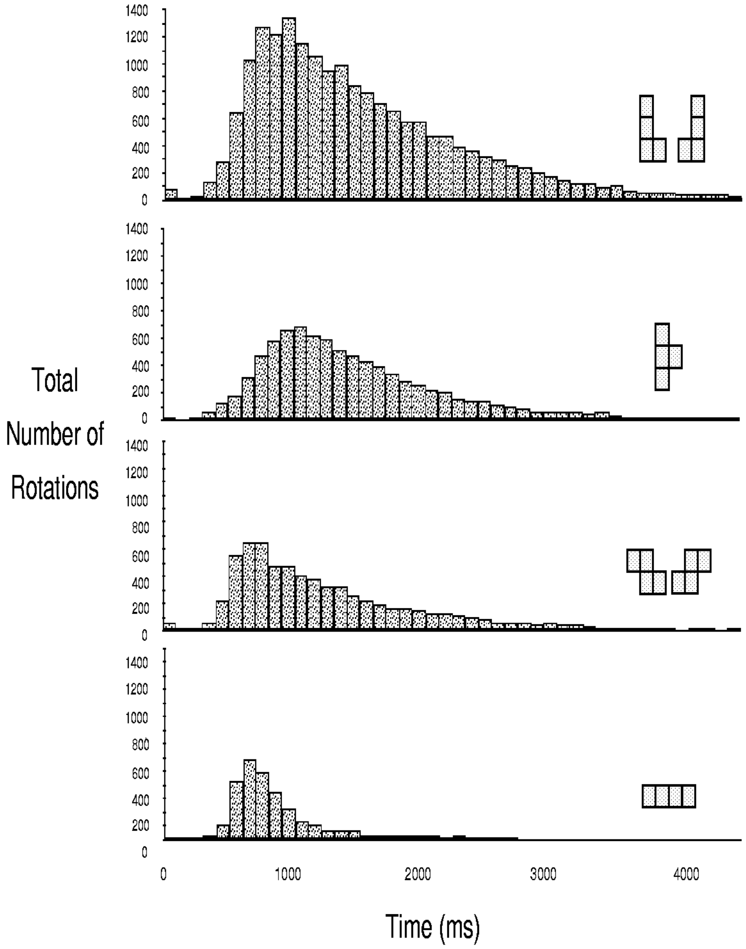


Figure 7. These histograms show the time-course of rotations for ’s, ’s, ’s, and ’s. Each bin contains the total number of rotations performed within its time-window. Note that rotation begins in earnest by 400-600 ms, and on occasion, at the very outset of an episode. The implication is that planning cannot be completed before rotation begins.

then, presumably, he or she ought to start out early toward that location and make corrections to zoid orientation as plan revisions are formulated. Early execution, on average, ought to save time.

In theory, such an account is plausible. That is, we would expect to find extra rotations in interleaving planners because the earlier an estimate is made, the greater the chance it will be wrong, and hence the more likely the agent will make a *false start*.

In fact, however, given the time course and frequency of rotations we observe in Tetris, particularly among skilled players, an explanation in terms of false starts makes no sense. First, the theory does not explain why an agent might start executing before having *any* estimate of the final orientation of a zoid. We have observed that occasionally a zoid will be rotated very early (before 100 ms), well before we would expect an agent to have any good idea of where to place the zoid. This is particularly clear given that at 100 ms, the zoid is not yet completely in view, and sometimes the agent cannot even reliably guess the zoid's shape.[2] Since Phase Two has barely begun, it is hardly reasonable that Phase Four is producing an output that an agent ought to act on.

Second, there is a significant cost to a false start even when the agent has reasonable grounds for an estimate. When a zoid is rotated beyond its target orientation, the agent can recover only by rotating another one to three more times, depending on the type of zoid. The time required to recover will depend on how long it takes to physically rotate the zoid. The shortest time between keystrokes in our data is about 75 ms, and the average time between keystrokes is around 250 ms. Thus, if the fastest a player can physically rotate is near the shortest interkeystroke interval, and the average time to rotate is around the average interkeystroke interval, then recovery time for a false start is between 75 ms and 750 ms. In the average case, this is a significant price to pay unless false starts are uncommon. As noted, however, extra rotations are regularly performed, even by experts. Apparently, players are bad at estimating final orientations early. But then why should they act on those estimates? If the probable benefit of rotating before finalizing a plan is low, it is better to wait for a more reliable estimate than to act incorrectly and have to recover. In this case, interleaving seems like a bad strategy for

[2] A following section, "Early Rotations for Discovery", specifically discusses what can and cannot be known about a zoid's shape at an early stage in an episode.

a well-adapted agent.

In our view, the failure of classical and interleaving planners to explain the data of extra rotations is a direct consequence of the assumption that the point of action is always pragmatic: that the only reason to act is for advancement in the physical world. This creates an undesirable separation between action and cognition. If one's theory of the agent assumes that thinking precedes action, and that, at best, action can lead one to re-evaluate one's conclusions, then action can never be undertaken *in order to alter the way cognition proceeds.* The actions controlled by Phase Four can never be for the sake of improving the decision-making occurring in Phase Three, or for improving the representation being constructed in Phase Two. On this view, cognition is logically prior: cognition is necessary for intelligent action, but action is never necessary for intelligent cognition.

To correct this one-sided view, we need to recognize that often the point of an action is to put one in a better position to compute more effectively: to more quickly identify the current situation; to more quickly retrieve relevant information; to more effectively compute one's goal. For instance, if the action of rotating a zoid can actually help decision-making—so that it is easier to compute the goal placement after the rotation than it is before—it suddenly makes sense to interleave action and planning. The time it takes to perform one or two rotations can more than pay for itself by improving the quality of decisions.

To make our positive case compelling we turn now to the interpretation of data we have collected on rotations and translations. How does adding the category of epistemic action show extra rotation and translation to be adaptive for good players? We consider rotation first.

EPISTEMIC USES OF ROTATION

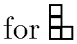
Pragmatically, the function of rotation is to orient a zoid. We speculate that rotation may serve several other functions more integral to cognition. Principally, rotation may be used to:

1. unearth new information very early in the game,
2. save mental rotation effort,
3. facilitate retrieval of zoids from memory,

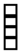
4. make it easier to identify a zoid's type,
5. simplify the process of matching zoid and contour.

Each of these epistemic actions serves to reduce the space, time, or unreliability of the computations occurring in one or another phase of Tetris-cognition. We are not claiming, however, that every player exploits the full epistemic potential of rotation. From a methodological standpoint, it is often hard to prove that an agent performs a particular action for epistemic rather than for pragmatic reasons because an action can serve both epistemic and pragmatic purposes simultaneously. Rotating a zoid in the direction needed for final placement may also help the player identify the zoid. This frequently makes it difficult to quantify the relative influence of epistemic and pragmatic functions. Nonetheless, the two functions are logically distinguishable, and there are clear cases in which the only plausible rationale for a particular choice of action is epistemic.

Early Rotations for Discovery

When a zoid first enters at the top of the screen, only a fraction of its total form is visible. At medium speed, a zoid descends at the rate of one square every 150 ms. Therefore, it takes about a half second for  's full image, for instance, to emerge. It is clearly in the interest of a player to identify the complete shape as soon as possible. This is easly done when only one type of zoid is consistent with a partial image. But in general, the emerging partial image could be produced by many zoids; it is ambiguous.

Given the value of early shape recognition, we would predict that if a strategy exists for disambiguating shapes early, then good players would strike on it. And indeed they have. By rotating an emerging zoid, players can expose its hidden parts, thereby uncovering its complete visual image 150 to 300 ms earlier than if they waited for it to appear naturally.

Sometimes early rotation is not necessary if a player has perfect knowledge of where shapes emerge. For instance, emerges in column 4, and  emerges in column 5 (see Figure 8). Let us say that an emerging zoid is ambiguous in shape but not position if there are other zoids which produce partial images that look just like it, but in different columns. If the early images are identical—i.e., in both image and column—we say the emerging zoid is ambiguous in both shape and position. A zoid that is ambiguous in

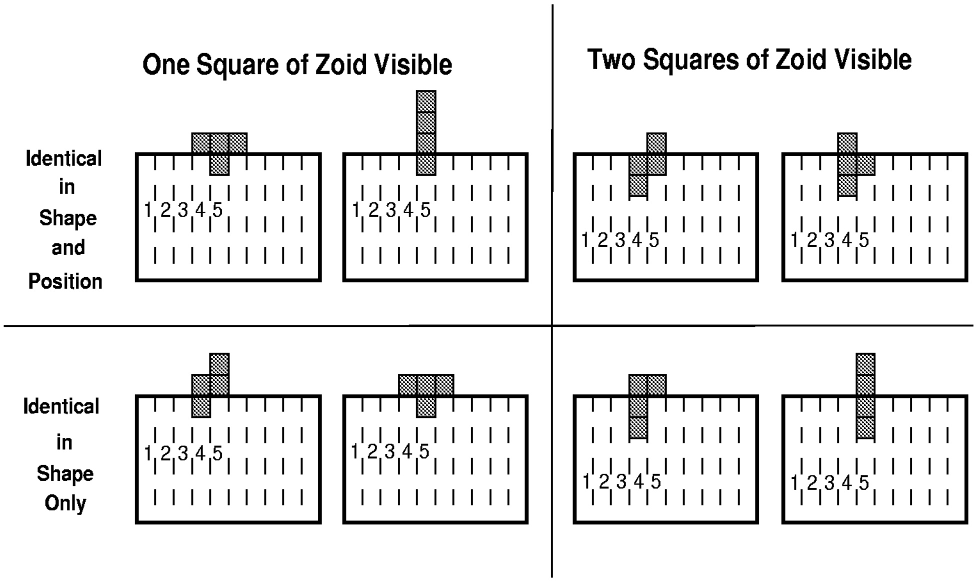

One Square of Zoid Visible
Two Squares of Zoid Visible
Identical in Shape and Position
Identical in Shape Only
1 2 3 4 5

Figure 8. This figure shows zoids as they first emerge at the top of the screen. To the left, they are one square in, and to the right, two squares in. At the top, the visible portions of the zoids are identical both in position and in shape. At the bottom, zoids are identical in shape alone; careful examination reveals that the images are in different columns. Players have a much greater tendency to rotate partially hidden zoids ambiguous in both shape and position than they have of rotating partially hidden zoids that are ambiguous in shape alone.

both shape and position produces an early image such that no matter how much a player knows, it is impossible to tell which zoid is present solely on the basis of the early image.

Our data show that a player is more likely to rotate a partially hidden zoid that is ambiguous in both shape and position than one ambiguous in shape alone. Partially hidden zoids ambiguous in shape only are *not* rotated more than completely unambiguous ones.

This suggests that players are sensitive to information about column because, in principle, zoids ambiguous in shape alone are distinguishable by column. Hence early rotation would add no new information. Yet, when interviewed, no player reported noticing that zoids begin falling in different columns. Thus, although players are sensitive to column, and are more likely to rotate in those cases where it is truly informative to do so, they do not realize they have this knowledge.

Early rotation is a clear example of an epistemic action. Nonetheless, one might try arguing against this view by suggesting that there is pragmatic value in orienting the zoid early, and so its epistemic function is not decisive. Such an explanation, however, fails to explain why partial displays that are ambiguous in shape and position are rotated more often those that are not ambiguous shape and position. Nor would such an explanation make sense if we believe that an agent has yet to formulate a target *orientation* for a zoid at this early stage. It is certainly possible that a player begins an episode with a set of target spots on the board where he or she would like to place the current zoid. Some players do report having hot spots in mind before an episode begins. And some of these players do *translate* a zoid early on the assumption that whatever shape emerges, they are likely to want to place it in a hot spot. But such early intentions explain early translation, not early rotation. If one does not know the shape of a zoid, there is no sense in rotating it to put it in the right orientation. Accordingly, it is hard to escape the simple account that the point of early rotation is to discover information normally available later, and that the benefits of performing this action outweigh the cost of potentially rotating a zoid beyond its eventual goal orientation. Competing pragmatic explanations are simply not as plausible as epistemic ones.

We have just considered how rotation may aid in early encoding, i.e., in Phase Two. In Phase Three—the decision phase—rotation also serves a variety of epistemic functions.

Rotating to Save Effort in Mental Rotation and Mental Imagery

In Phase Three, players determine where to put the zoid. They must have a useful representation of the currently falling shape, and a useful representation of the contour (or segments of the contour) to compare or match to find an appropriate placement. In our brief characterization of Phase Three above, we described the heart of the process as a search for the largest uninterrupted contour segment that the zoid can fit. This process probably involves matching chunks.

At least two versions of this comparison process can be distinguished.

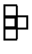
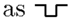
Method One: The player identifies the type of the zoid before looking for possible placements, using knowledge of all orientations to search for snug fits. This means that the player extracts an abstract, *orientation-independent* description of the shape, or chunk, before checking for good placements.

Method Two: The player does not bother to compute an orientation-independent representation of the zoid or chunk. Leaving the representation in its orientation-sensitive form, the player redirects attention to the contour, looking for possible matches with the orientation-specific chunk. In this second method, contour checking can begin earlier than in the first method, but to be complete, the process of contour checking must be repeated for the same zoid or chunk in all its different orientations. Needless to say, we may discover players who use some of each method, possibly with the two running concurrently.

When we look more closely at these methods, we see several points where epistemic actions would be useful.

Consider method two first. Somehow a player must compare the shape of a zoid in all its possible orientations to fragments of the contour. To do this, the player may compare the zoid in its current orientation to the contour, then use *mental imagery* to recreate how the shape would look if rotated (see Figure 9).[3] Another possibility—far more efficient in its use of time—is that the player may rotate the zoid physically and make a simple, orientation-specific comparison.

The clearest reason to doubt that deciding where to place a zoid involves mental rotation is that zoids can be *physically* rotated 90° in as few as 100 ms, whereas we estimate that it takes in the neighborhood of 800 to 1200 ms to mentally rotate a zoid 90°, based on pilot data such as that displayed in Figure 10.[4] We obtained these data using a mental rotation task very

[3] Possibly, the player may use pattern recognition, feature matching, or case-based reasoning to judge how well the zoid fits, even in other orientations, but the judgement is made on the basis of the zoid in its current orientation. For instance, the player may know on the basis of past cases that  fits in a contour segment such as  when rotated into . We ignore this method here. It gives rise to its own set of epistemic actions we will not consider.

[4] This comparison may be slightly misleading if we assume that it takes less time to mentally rotate a zoid a second time than it does to rotate it the first time, owing to a self-priming effect. But, given how large the disparity between physical and mental rotation is, we still expect a significant difference in favor of physical rotation.

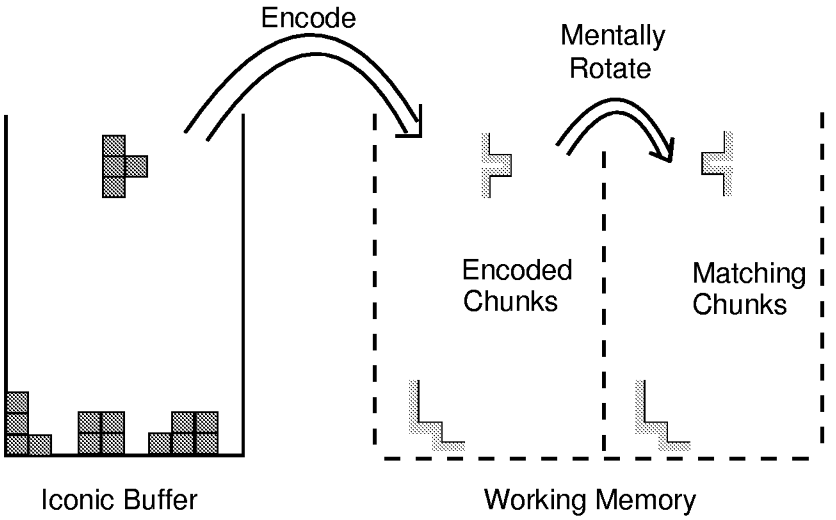
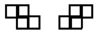
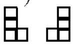


Figure 9. A chunk extracted from the image of a zoid is normalized by internal processes and compared to a chunk extracted from the image of a contour. A computationally less intensive technique of comparing zoid and contour would rely on physical rotation of the zoid to take the place of the internal normalization processes.

similar to the one used by Shepard and Metzler (1971). In our experiment, two zoids, either S-shaped () or L-shaped (), were displayed side-by-side on a computer screen. The zoids in these pairs could differ in orientation as well as handedness, but in all cases, both items were of the same type. To indicate whether the two zoids matched or whether they were mirror images, subjects pressed one of two buttons. Three Tetris players participated: one intermediate, one advanced, and one expert. Each subject saw eight presentations of each possible pair of zoids. The results, as graphed in Figure 10, show reaction time as an increasing function of the angular difference between the orientations of the two zoids (from 0° to 180°).

Even allowing an extra 200 ms for subjects to select the rotate button, the time saving benefits of physical over mental rotation are obvious. But time is not all that is saved. There are also costs associated with the attention and memory needed to create and sustain mental images (Kosslyn, 1990). For instance, suppose that matching proceeds by comparing rotated chunks of a zoid with chunks of the contour. Even if chunk rotation and comparison is faster than we expect, there are still significant memory costs to maintaining a record of the chunks that have already been checked. The generate and test process requires repeatedly consulting the zoid image and selecting a new chunk to check. The net result is that the visuo-spatial (Baddeley, 1990) would soon fill up with a) re-oriented zoid chunks, b) the contour chunk that is the target for matching, c) some record of the zoid chunks already tested, and d) a marker of where on the contour the current contour chunk comes from. It seems far less demanding of visual memory to simply do away with the extra step of normalizing (i.e., rotating) zoids or chunks of zoids, and compare zoid chunks to contour chunks directly. Hence, pending a deeper account of the process, it seems obvious to us that physically rotating is computationally less demanding than mentally rotating.

The same conclusion applies even if players use method one for generating candidate placement locations.

Rotating to Help Create an Orientation-Independent Representation

In method one, players extract an abstract, *orientation-independent* description of the zoid or chunk before checking the contour. They are willing to pay the processing price of extracting this abstract representation because

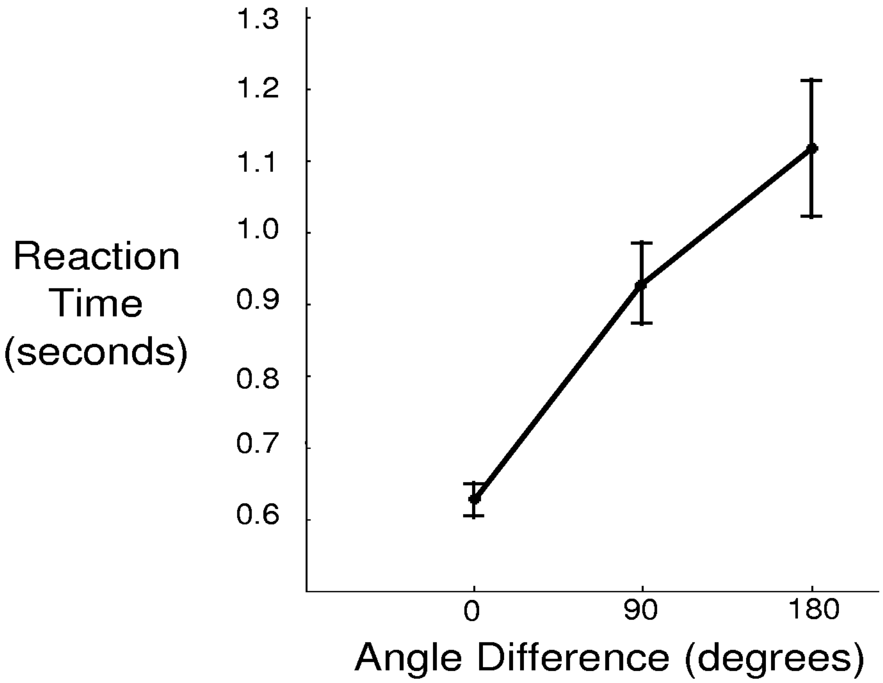


Figure 10. This graph shows the results of a pilot study on the mental rotation of Tetris shapes by players of differing skill levels. Reaction time (in seconds) is plotted against difference in orientation of two displayed L-shaped zoids (only differences from 0° to 180° are plotted). Only correct "same zoid" answers are included; i.e., conditions in which both zoids were either of type or of type . A linear relationship between reaction time and angle-difference is readily apparent. The error bars represent 95% confidence intervals.

once they have an orientation-independent representation of a zoid, it is not necessary to rotate the zoid further to test for matches. Nonetheless, external rotation is still epistemically useful because it is helpful in constructing orientation-independent representations in the first place.

What does it mean to have an orientation-independent representation? From an experimental perspective, it means that it should take no more time to judge whether two shapes are the same, however many degrees apart the two have been rotated. Players' reaction times on mental rotation tests should be plotted as a horizontal line, rather than the upwardly sloping line we see in Figure 10. Total reaction time should be the sum of the time needed to abstractly encode the first shape (presentation), the time to abstractly encode the second shape (presentation), and the time to compare the abstract encodings. Moreover, we would expect that both the time to abstractly encode different presentations, and the time to compare abstract encodings should be constant across all trials.

We have not observed flat line performance on mental rotation tests of very experienced players, so we must be skeptical of the hypothesis that players use abstract orientation independent representations.[5] But in other studies of extremely practiced shape rotaters, it has been found that, in fact, the more exposure subjects have to shapes in test orientations, the closer to flat line performance they display. (Tarr & Pinker, 1989) The explanation Tarr and Pinker offer is that with practice subjects begin to acquire a *multiple-perspective* representation of the shape. In the context of Tetris this means that if experts exhibit flat line performance on rotation tests, then we should expect them to have built up multiple representations of the zoids. Determining the type of a zoid would involve *activating* a set of representations of the zoid, in which the internal images of each of its orientations is strongly primed—so strongly primed that any one could be retrieved more quickly than it could be generated by mental rotation.

Contrary to our current expectations, if players do create multiple-

[5] It is quite possible, however, that it takes longer to create an orientation independent representation than to rotate an image just once. In that case, Tetris players may rotate mental images in rotation tests, but find it worthwhile to pay the fixed costs of constructing an orientation independent representation if they know they will be facing repeated judgements concerning the same shape. Indeed, players may automatically abstract an orientation independent representation, if they see a piece long enough, so the failure to display flat line performance may be an artifact of the standard experimental design.

perspective representations, external rotation could play a valuable role in speeding up the multiple-perspective encoding process. Consider what it means, from a computational perspective, to activate (or encode) a multiple-perspective representation. Presumably, the agent enters a state in which the complete set of orientation specific representations are active, or at least, strongly primed. The process by which this activation takes place is identical to retrieval. Thus, each image of a shape serves as an index, or retrieval cue, for the multiple-perspective representation.

How might physical rotation help such a retrieval process? One conjecture, which is ripe for experimental testing, is that retrieval is faster the more environmental support there is (Park & Shaw, 1992). For instance, we speculate that it takes less time to complete a retrieval using $n+1$ indices than to complete a retrieval using n indices. Thus, we might expect that if it takes a subject a total of 1200 ms to identify which type of L-shaped zoid is present when shown a single token, such as , it may take less time, say 1000 ms, to identify the type if shown more than one token: for instance, for 600 ms immediately followed by for 400 ms. Rapid presentation of different perspectives of a zoid might stimulate faster retrieval than presentation of a single perspective.

In an attractor space model of retrieval—for instance, in a Boltzman machine—this is exactly what we would predict. Consider a word completion task in which any three letters are sufficient to uniquely identify a target word. Given a stimulus such as **c * t * r * ***, and a set of legal words in which **catarrh** is the only valid completion, the time for the machine to settle on the correct target will be some finite value t.[6] We assume that if the machine is shown a second stimulus consistent with the first but with three different letters filled in, e.g., *** a * a * r ***, it will settle more quickly, say $t-a$. The first stimulus starts the system near the top of the energy sink that represents the target word, and the second stimulus pushes the system deeper down the well.[7]

In a Boltzman machine model of activation, then, rotation will serve

[6] The actual time a machine takes to settle on the correct target is, of course, implementation-dependent.

[7] To be sure, a Boltzman machine may not *always* settle more quickly in this case. The topology of the energy surface and the relative informativeness of each of the two cues (among other factors) could also be important in determining how long it takes to find the right attractor.

the useful function of speeding up the activation process. In this case, *two cues are better than one.* Because rotation is the means of generating the second cue, and rotation is quick enough to save time in the settling process, it can play an epistemically valuable role.

Rotating to Help Identify Zoids

It is an open question whether agents use multiple perspective representations of zoids (or chunks). It is not an open question whether there is a phase where zoids are first represented in their current perspective as particular zoid shapes (or chunks of zoids). On our account, the process by which particular zoids are encoded in working memory has three logical steps. In the first, simple features such as lines, corners, and colors are extracted from the image; in the second, orientation-specific corners and lines—conjunctive features of the image—are extracted; and in the third step, structured sets of conjuntive features—perceptual chunks—are identified and encoded explicitly in working memory. Both steps two and three require attention. It is reasonable to suppose, then, that fast perceptual chunking is the result of a highly trained attentional system, and that any improvement in chunking is due to improvement in the attentional strategy controlling chunk and zoid recognition. Thus, we hypothesize that when subjects improve at identifying chunks and zoids, it is because they have learned to better attend to simple features represented in the iconic buffer.[8]

We can recast this hypothesis in a more computational form: we can say that the more expert a player, the more efficient he or she ought to be at searching for the features which indicate the presence of specific zoids or chunks. Accordingly, one way to represent the extra competence of experts is in terms of the optimality of a decision-tree for finding chunks or zoids by means of queries directed at the iconic display. In decision theory, a decision-tree is deemed optimal when the most informative question is asked first, followed by the next most informative question and so on. If the iconic buffer is a matrix of cells—and encodes no more than a single zoid, or a single contour fragment at a time—the optimal decision-tree will consult the

[8] The argument to be presented applies equally well whether it is the iconic buffer , or early representations in the visuo-spatial working memory, which is probed by attention. The crucial factor is that the features attended to are tied to an egocentric coordinate system rather than to an object-centered one.

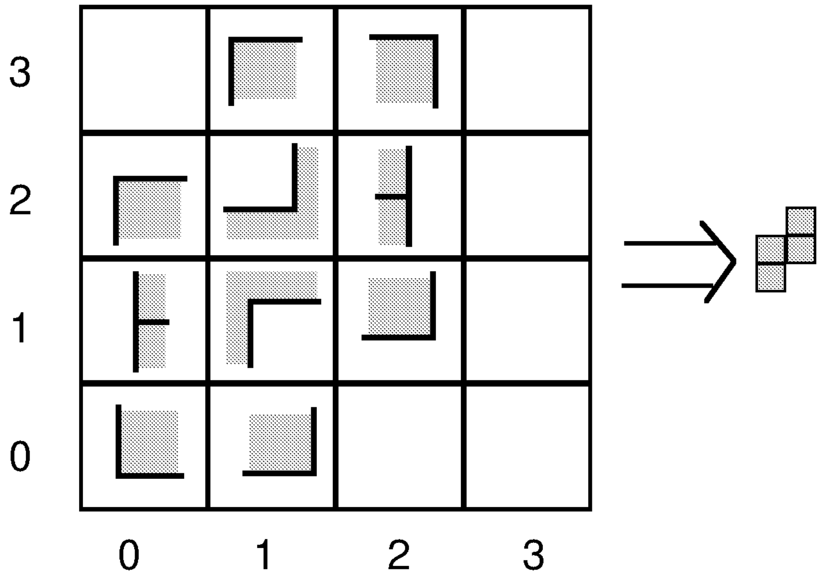


Figure 11. The iconic buffer is a 4 × 4 matrix of cells, each of which may contain a primitive feature.

minimal number of cells to reliably extrapolate to the contents of the whole matrix (see Figure 11).

Given the shape of tetrazoids, experts may sometimes rotate zoids because, if encoding operates by a mechanism at all like a decision tree, then *rotating can be an effective way of reducing the number of attentional probes needed to identify a zoid.* Compare Figures 12 and 13. The decision tree in Figure 12 assumes the expert identifies the zoid without rotating it. As can be seen, if the expert first examines cell $(1, 1)$, then, a decision will require either one, two, or three questions directed at the matrix to identify the zoid, depending, of course, on the zoid present and the contents of $(1, 1)$. The decision tree in Figure 13, however, shows that if the agent can also rotate the zoid between its attentional probes of the matrix, an identification can be made in at most two questions. Thus, rotation can be used to streamline the program controlling attention. An expert can operate with a smaller decision-tree if rotation is included in the set of actions the tree can call on.

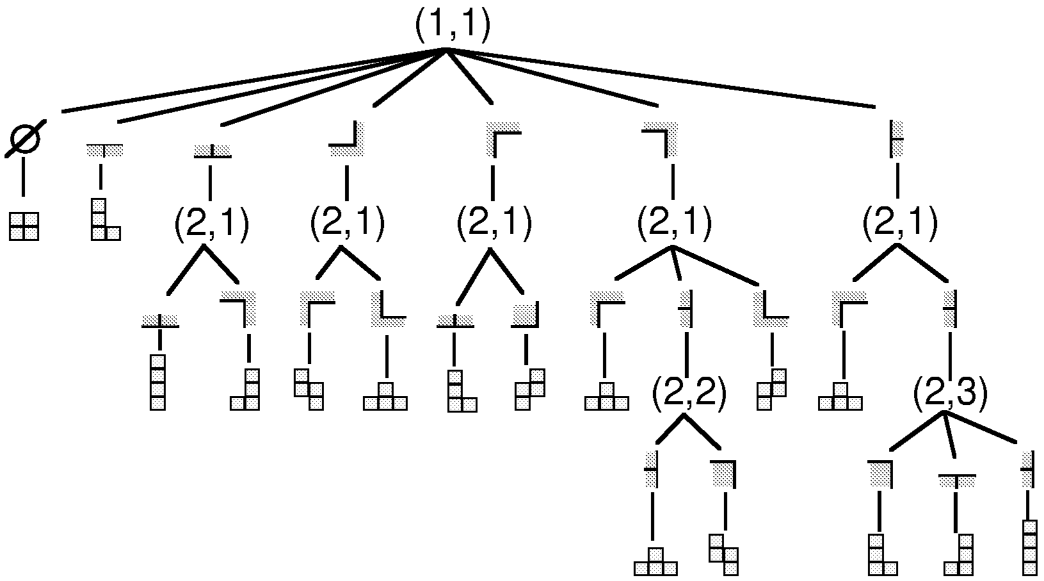


Figure 12. This decision-tree directs a series of questions at specific cells in the iconic buffer in order to identify what type of zoid is present. The tree first probes cell $(1,1)$. If the buffer is the one in Figure 11, cell $(2,1)$ is queried next, leading to the identification of ⊞ .

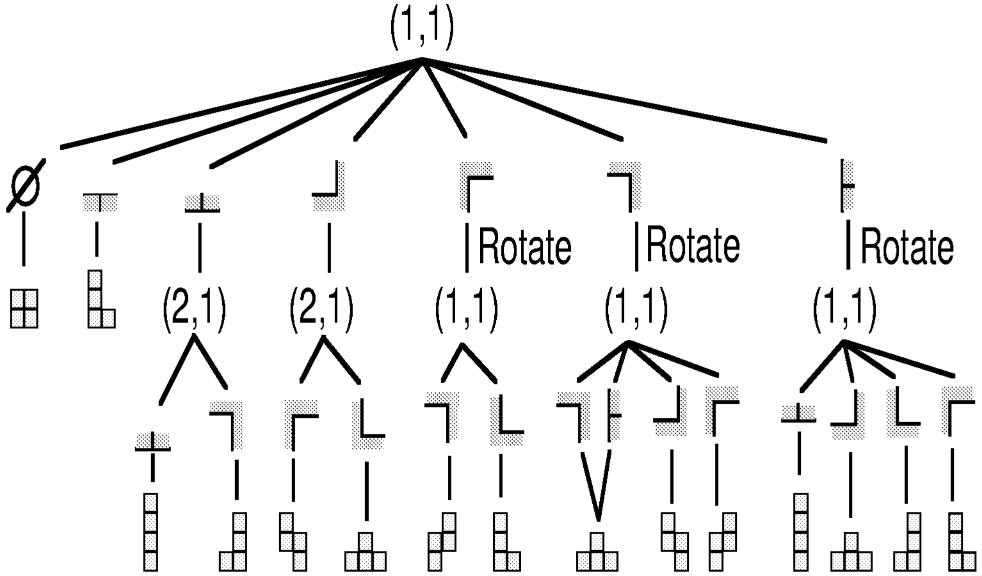


Figure 13. If the decision-tree incorporates calls to external rotation operations, its maximum depth is two. In addition, attention need not shift from cell $(1,1)$ most of the time.

But this may be only part of the story. So far, we have argued that identification involves domain-specific control of attention, and that extra rotations may be a side effect of a streamlined program regulating this control. A second reason experts may make superfluous rotations is that, paradoxically, it is the lazy thing to do. Although we do not know if it takes less energy on the part of an attention mechanism to consult the same cell twice, it is possible that a lazy attention mechanism might prefer to re-ask for the value of a cell, rather than focus on a new cell. This is an obvious strategy when new data has just arrived because change is automatically interesting to the nervous system. This idea of finding a strategy that minimzes the number of cells probed makes sense in a decision-tree account of attention as long as it costs less to consult the same cell on successive inquiries. In that case, the decision-tree in Figure 13 would be preferred over the decision-tree in Figure 12 because probing the same cell on most of the successive queries would put less strain on the attentional system.

The implication of both arguments, we believe, is that it is adaptive to build attentional mechanisms that are closely coupled with actions such as rotation. The close coupling between attention and saccades is already accepted, why not extend this coupling to include more molar actions such as rotation?

Rotating to Facilitate Matching

So far we have assumed that matching is a primitive process in working memory: zoid chunk and contour chunk can be compared and matched only if they are explicitly represented in working memory. To make certain that enough chunks of different sizes are tested to guarantee finding the largest matching chunks, a player can rely on either externally rotating a zoid, mentally rotating a zoid, or mentally accessing a multiple-perspective representation of a zoid to generate as many candidate chunks as time will allow.

Are we justified in assuming that matching occurs in working memory? And that symbolic matching, primitive or not, is really the fastest way of determining a fit between a zoid fragment and a contour fragment?

An alternative possibility is that matching is a perceptual process. The general idea is simple enough. Matching requires noting the congruence of two structures. If the structures are simple—such as lines or rectangles lying in the same orientation—it may be possible to note their congruence by using some attention-directed process such as a visual routine (Ullman, 1985) applied directly to the early bitmap-like representation. In that case, matching might actually be an element of Phase Two—the phase in which salient features of the situation are extracted and encoded—instead of an element of Phase Three—the phase in which operations are applied to structures in working memory.

External rotation plays a role in this alternative story because because we have to explain how new candidate zoids or zoid chunks are generated. Because we are considering a mechanism in which matching occurs very early, there muct also be a mechanism for generating candidates very early. The only certain way to get information about new candidates into the iconic buffer is through perception. It is possible, of course, that new zoid orientations may be generated through mental rotation. But, first, it is not known whether mental imagery can create bitmaps, or whether it affects only representations in working memory—as in Baddely's visuo-spatial sketchpad, for instance. Second, if mental rotation does modify the pre-attentive iconic buffer—where the bitmaps reside—players would probably prefer to create the relevant bitmaps by external rotation rather than by mental rotation because, as mentioned earlier, external rotation is faster. And third, it is likely that physical rotation is less cognitively demanding than mental rotation. Iconic memory needs to be refreshed every 200 ms (Reeves & Sperling, 1986). Thus, if a player uses mental imagery to flood the iconic buffer, he or she will have to refresh the buffer every 200 ms. It is much easier to generate tokens by bringing them in through the visual system than by internally creating them. Therefore, even if matching operates by perceptually noticing correspondence, we have another reason for preferring external rotation both to mental rotation and to multiple-perspective representations.

So ends our account of the epistemic uses of rotation. We conclude our discussion of the data with a brief description of one epistemic use of translation.

TRANSLATION AS AN EPISTEMIC ACTION

The pragmatic function of translation is to shift a zoid either right or left to permit placement in an arbitrary column. Translation usually serves this pragmatic purpose. But we have found at least one unambiguously epistemic use of translation: to verify judgement of the column of a zoid. In about 1% of the cases when a player drops a zoid, the act of dropping is preceded by a behavioral routine of translating the zoid to the wall and then back again. See Figure 14. Because the accuracy of judging spatial relationships between visually presented stimuli varies with the distance between the stimuli (Jolicoeur, Ullman & Mackay, 1991), a zoid dropped from a height of 15 squares has a greater chance of landing in a mistaken column than a zoid dropped from a height of three squares. Thus, the obvious function of this translate-to-wall routine is to verify the column of the zoid. By quickly moving the zoid to the wall and counting out the number of squares to the intended column, a player can reduce the probability of a mishap.

As epistemic actions go, this one is hard to confuse with a pragmatic action. By definition, it requires moving the zoid away from the currently intended column, and hence it cannot be a pragmatically good move. More-

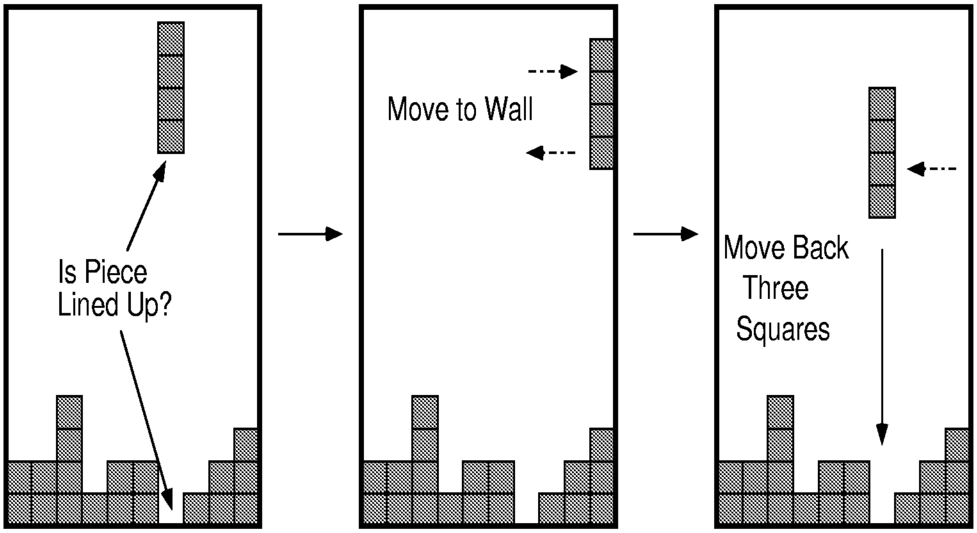


Figure 14. In a small percentage of cases players will drop certain zoids only after translating them to the nearest wall and then back again, as if to verify the column of placement. In this figure, ▯ is translated to the outer wall and back again before it is dropped. The explanation we prefer is that the subject confirms that the column of the zoid is correct, relative to his or her intended placement, by quickly moving the zoid to the wall and simultaneously counting and tapping out the number of squares to the intended column.

Table 1
Ordinary Drop Distance vs. Translate-to-Wall-then-Drop Distance

	Intermediate	Advanced	Expert
Mean Drop Distance	13.18	13.69	15.65
Mean Drop Distance after Translate Routine	19.04	19.33	20.05

Note. Within each skill level, the two means differ significantly as judged by a t test with $\alpha = .05$.

over, it cannot sensibly be viewed as a mistaken pragmatic action because the procedure is more likely to occur the higher the drop. As shown in Table 1, experts drop a zoid, on average, when it is about 13 squares from its resting position. On those occasions when they also perform the translate-to-wall routine, the zoid is dropped, on average, from about 19 squares above its resting position, 6 squares higher than usual. The only reasonable account for this regularity is that the higher the zoid, the more the player needs to verify the column. Moreover, as shown in Figure 15, the greater the drop distance, the more likely the drop is verified using the translate-to-wall routine. At great heights above the zoid's resting position, the pragmatic cost of moving away from the goal column is more than offset by the epistemic benefit of reducing possible error.

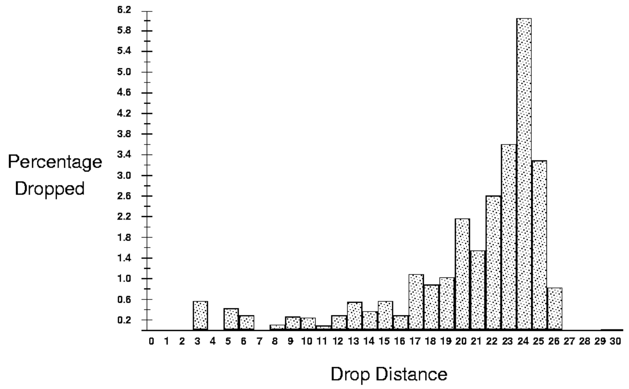


Figure 15. This graph plots the percentage of dropped zoids that followed a translate-to-wall routine against the distance they were dropped. The higher the drop, the more likely it followed a verification routine.

DISCUSSION

To explain our data on the timing and frequency of rotations and translations regularly performed by Tetris players, we have argued it is necessary to advert to a new category of action: epistemic actions. Such actions are not performed to advance a player to a better state in the external task environment, but rather to advance the player to a better state in his or her internal, cognitive environment. Epistemic actions are actions designed to change the input to an agent's information-processing system. They are ways an agent has of modifying the external environment to provide crucial bits of information just when they are needed most.

The processing model this suggests to us is a significant departure from classical theories of action. Its chief novelty lies in allowing individual functional units inside the agent to be in closed-loop interaction with the outside world. Figure 16 graphically depicts this tighter coupling between internal and external processes. As in the cascade model mentioned previously, processing starts in each phase before it is complete in the prior phase. But in this case, the output of Phase Two can bypass Phase Three and Phase Four, activating a motor response directly. Similarly, individual components of Phase Three can bypass Phase Four.

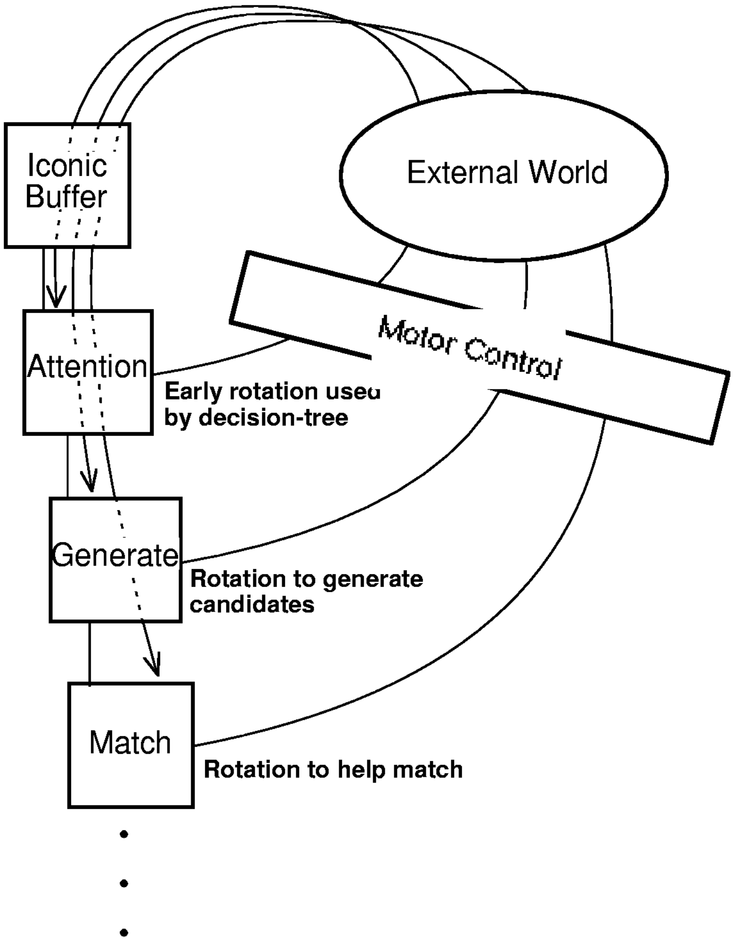


Figure 16. In this model, calls for rotation from attentional processes, or from candidate generation processes, cause changes in the world which feed back into those very processes. Because of the tight coupling between action and what is perceived, the fastest way to modify the informational state of an internal process may be to modify its next input.

To return to an example already discussed, suppose attention operates as if driven by a decision-tree. The attentional system may request rotations in the same way that it requests directing attention to cell (i, j) in the iconic buffer. These requests are not sent to the Phase Three processes operating on working memory, as if to be approved by a higher court. They are temporary, time-critical requests which have no bearing on the pragmatic choice of where to ultimately move. The point of the request is very specific: to cash in on the speed at which input can be changed. If a change of input will help complete the computations that constitute selective attention faster than the attention system can compute on its own, it would be adaptive to link attention directly to certain simple motor actions.

The property of Tetris that makes such a strategy pay off is that the local effects of an action are totally determinate. There are no hidden states, exogenous influences, or other agents to change the result of hitting the rotate key. There is a dependable and simple link between motor action and the change in stimulus. Consequently, a well-adapted attentional mechanism might incorporate simple *calls to* the world as part of its processing strategy.

A similar story can be told for Phase Three, in which placements are generated and matched or tested for goodness. A well-timed rotation request can provide just the input needed to generate a new candidate, or to faciliate a match. Again, because of the tight coupling between action and local effect, the agent can count on input changing in the desired way. Because hitting the rotate key reliably changes what is perceived, this action can be relied on to help *think up* new possible matches.

One may object that postulating a link between such obviously distant processes as attention and motor control is ad hoc. How can processes concerned with attention, which, in our account, are responsible for extracting and encoding chunks from the iconic buffer, have a direct effect on motor control?

We have two replies. First, the analysis of Tetris-cognition into four phases with sparse interconnections is an idealization that has only partial neurophysiological basis. In the case of selective attention, there are a host of separate brain regions involved in encoding the bitmap features in the iconic buffer. Some of these have close connections to motor cortex (Felleman & Van Essen, 1991; Sereno & Allman, 1991). Certainly, it is not outrageous to suppose that there is motor involvement in selective attention since there already exists a close connection between attention and the oculomotor system responsible for saccades. Perhaps there is a similar connection between attention and highly trained key pressing responses.

Second, we can create a more complicated picture of the interrelations among processes involved in Tetris-playing than the one presented in Figure 2. Consider Figure 17, which displays a highly interconnected network of processes for attention, candidate generation, matching, and rotation. Obviously, this does not represent a strictly feedforward system: there are backward links from **generate** candidates and **match** to **attention**, as well as from all three to **motor arbitrate**. We have already discussed how **match** and **rotate** can benefit from sending requests back to **attention**. In the same way, candidate generation can benefit from sending requests back to attention because the process of generating new candidate placements requires trying out new zoid chunks and new contour chunks, and an easy way to create such chunks is by looking at zoid and contour anew. The one complication this connection scheme adds to the process is that requests for motor actions must be arbitrated, hence the addition of the **motor arbitrate** process. This kind of model follows the distributed framework proposed by Minsky (1986).

If this way of thinking has merit, it suggests that we begin asking additional questions when studying behavior. For instance, we should now confront a task and ask not only, "How does an agent think about this task, e.g., categorize elements in it, construct a problem space representation for it?" but also, "What actions can an agent perform that will make the task more *manageable*, easier to compute?"

This represents a shift from orthodox cognitivist approaches. A central theme in cognitive psychology has been to discover the organizing principles agents use to structure their environments. One way to study this is to vary properties of the stimuli agents find in their environment, and to observe the effects of these changes on such performance criteria as time to recall, recognize, complete, and so on. How does context affect performance? If elements of the stimulus are grouped one way rather than another, are they better, faster, more often recalled? What serves to distract or to enhance recall and recognition? A noteworthy aspect of this method is that the subject, in important respects, has little control over the stimulus. The experimenter varies the external stimuli with the hope of discovering the subject's internal organizing processes.

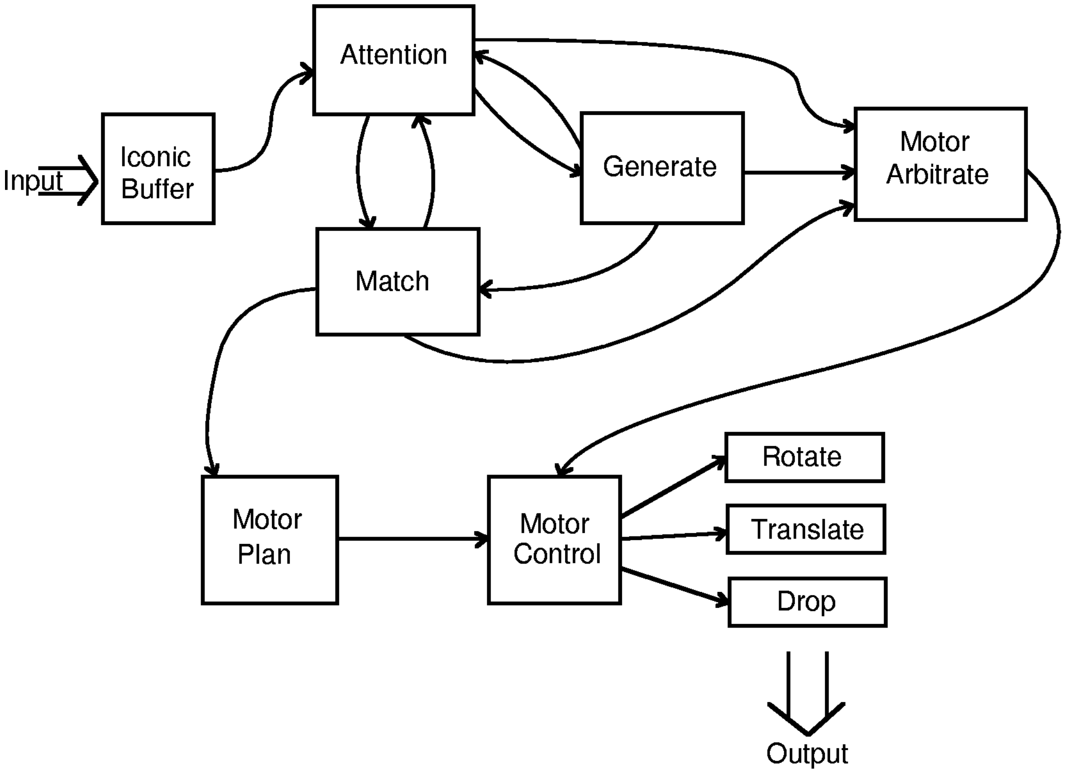


Figure 17. A more complicated model of the processes occuring in Tetris-cognition would represent particular functional parts as a directed network of mental processes able to pass messages between each other. The only significant deviation from the sketch in Figure 16, is that two way links between attention, candidate generation, and matching are shown, and a new process, called an arbitrator, is introduced to intervene between the possible calls to translate, rotate, and drop.

There is, of course, nothing wrong with this approach. It permits controlled study. But it reflects a bias that the type of environmental structuring relevant to problem solving, planning, and choice, as well as to recall and recognition, *occurs primarily inside the agent.* That is, the environmental structure that matters to cognition is the structure the agent *represents* (or at least, presupposes in the way it manipulates its representations). No allowance is made for *offloading* structure to the world, or for arranging things so that the world pre-empts the need for certain representations, or pre-empts the need for making certain inferences. This leaves the performance of such pre-emptive and offloading actions mysterious.

To take a simple example, a novice chess player usually finds it helpful to physically move a chess piece when thinking about possible consequences. Why is this? From a problem-space perspective, the action seems totally superfluous. It cannot materially alter the current choices and considerations. Yet, as we know, by physically altering the board, rather than by merely imagining moving a piece, novices find it easier to detect replies, counter-replies, and positions. In like style, a slightly more advanced player often finds it helpful to change his or her spatial position, leaving the game intact but moving to a new vantage point to see if otherwise unnoticed possibilites leap into focus, or to help break any mind-set that comes from a particular way of viewing the board. In problem solving, it can be valuable to shake up one's presuppositions, to perturb the world to force the re-evaluation of assumptions—of preparatory set, to use the Gestaltists' term.

If the function of a particular action is as non-transparent as to jog memory, to shatter presuppositions, or to hasten recognition, then the agent's relation to the world is far more complex than usual psychological models suggest. No longer is choice the outcome of a simple two stroke engine—classify stimulus then select external response—or three stroke engine—classify stimulus, predict and weigh expected utility of responses, select external response. For the stimulus, in these epistemic cases, is not reacted to as an indicator of the state of the task environment, it is used as a reminder to do X, a cue that helps one to recall Y, a hint that things are not as once thought, or as a revision of input so that an internal process can complete faster. To make an analogy, just as the function of a sentence may be to warn, threaten, startle, promise, so the function of a perceived state may be to remind, alert, normalize, perturb and so on. *The point of taking certain actions, therefore, is not for the effect they have on the environment as much as for the effect* *they have on the agent.*

This way of thinking treats the agent as having a more *cooperative* and interactional relation with the world: the agent both adapts to the world as found, and changes the world, not just pragmatically, which is a first order change, but epistemically, so that the world becomes a place that is easier to adapt to. Consequently, we expect that a well-adapted agent ought to know how to strike a balance between internal and external computation. It ought to achieve an appropriate level of cooperation between internal organizing processes and external organizing processes so that, in the long run, less work is performed.

We conclude with a brief explanation of how accepting the category of epistemic action affects traditional AI planning.

Epistemic Actions and Theories of Planning

In the introduction, we suggested that AI planners might accomodate epistemic activity by operating in a state space whose nodes were pairs encoding both physical state and informational state. In that case, the payoffs a player receives from an action have two dimensions: a physical payoff, and an informational or epistemic payoff. The clearest examples of epistemic actions are those which deliver epistemic payoffs rather than pragmatic ones. The rationale, presumably, is that in each such case, after we have subtracted the cost of time lost performing the action, the expected epistemic or computational benefits still outweigh the expected net benefits of performing a pragmatic action.

The cost-benefit model that seems to apply here is one economists have used to characterize the tradeoff between information and action at least since Stigler's seminal paper "The economics of information" (1961). Stigler pointed out that for consumers with incomplete knowledge, say about the price of a camera, market information can be assigned a value by determining how much one could hope to save by shopping around before buying. If we assume that prices fit a normal distribution, the value of continuing to shop for a lower price decreases until an equilibrium is reached where the expected gain of one more inquiry is equal to its cost, the so-called "shoe leather cost".

In certain respects, the behavior of Tetris players conforms to this model. For instance, the probability that a player will rotate early is, to some degree, a function of the informativeness of the rotation. Early rotations are most informative when what is seen is ambiguous in both shape and position. The model also fits the translate-to-wall routine. Thus, we explain why the probability of translating to the wall and back before a drop varies with drop distance by pointing out that the greater the drop height, the more informative the verification and the less risky (costly) the action. It also explains why players physically rotate to save mental rotation: they can attain the same knowledge faster and with less effort than by mentally computing the image transformation. Rotating to facilitate matching has a favorable cost-benefit spread because matching via perception is fast, reliable, and uses less resources than matching in working memory.

The virtue of such a cost-benefit account is twofold. First, it permits us to continue modeling the decision about what to do next as a rational choice among accessible actions. Without a notion of epistemic payoff, we cannot justify why expert players sometimes choose pragmatically disadvantageous actions within a rational-agent calculus.

The second virtue of a cost-benefit account is that it partly explains the superior decision-making of experts over novices and intermediates. The more expert a player is, the more successful he or she should be in keeping the costs of computation down. Experts keep their costs lower, in part, by performing more epistemic actions.

But when we look more closely at what is involved in determining the epistemic payoff of an action, we see that simple economic models of costs and benefits fail because the benefits of an epistemic action depend in considerable detail on just what computations the agent is performing when undertaking an action. In classical decision analysis accounts (Howard, 1966; Raiffa, 1968), the value of a piece of information can be estimated by comparing the expected utility of an action after that information is discovered with its expected utility before. There is no need to know anything about the internal reasoning process of the agent to estimate how valuable that information gathering action ought to be. The same applies to Stigler's camera shopper. It is possible to determine the expected value of the next stop at a camera shop, assuming normal distribution, etc., quite independently of the shopper's reasoning process. We assume he or she is rational and remembers all previous prices. But when it comes to estimating the value of most of the epistemic actions we have discussed, it is not possible to ignore the particular cognitive processes they facilitate. Thus, the epistemic value of a rotation after 500 ms, say, will depend crucially on the current state of the agent, as well as on how it generates and tests candidate placements, and on how it attends to details of the contour and zoid. This requires understanding an agent's active cognitive processes to a level of detail unheard of in standard planning and rational decision accounts.

The upshot is that to incorporate epistemic actions into a planner's repetoire, we will need to cast aside the assumption that planning can proceed *without regard to* specific mechanisms of perception, attention, and reasoning. This idea is not foreign to the planning community, but to date it has been restrictively applied. For instance, in discussions of active vision—where repositioning sensors is a central concern—the decision about where to reposition a sensor is thought to depend on assumptions about the sensor's range, field of view, noise tolerance, and so on—all details about the inner functioning of the sensor. It is our belief that this need to know more about an agent's internal machinery generalizes to virtually all epistemic actions, and that once more is known about the internal machinery of action selection in particular domains, epistemic actions will emerge as far more prevalent than anyone would have guessed. We have argued for this view by showing how, in a game as pragmatically oriented as Tetris, agents perform actions that make it easier for them to attend, recognize, generate and test candidates, and improve execution. These actions make sense once we understand some of the processes involved in Tetris cognition. This same idea, we claim, holds generally throughout all of human activity.

References

Ambros-Ingerson, J. & Steel, S. (1987). Integrating planning, execution and monitoring. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 83–88.

Anderson, J. R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Baddeley, A. (1990). *Human memory: Theory and practice.* Boston, MA: Allyn and Bacon.

Bratman, M. (1987). *Intention, plans, and practical reason.* Cambridge, MA: Harvard University Press.

Chapman, D. (1989). Penguins can make cake. *AI Magazine*, *10*(4), 45–50.

Chernoff, H. & Moses, L. (1967). *Elementary decision theory.* New York: John Wiley.

Felleman, D. J. & Van Essen, D. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, *1*, 1–47.

Hitch, G. J. (1978). The role of short-term working memory in mental arithmetic. *Cognitive Psychology*, *10*, 302–323.

Howard, R. A. (1966). Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, *2*, 22–26.

Hutchins, E. (1990). The technology of team navigation. In J. Galegher, R. Kraut, & C. Egido (Eds.), *Intellectual teamwork: Social and technical bases of collaborative work.* Hillsdale, NJ: Lawrence Erlbaum.

Jolicoeur, P., Ullman, S., & Mackay, M. (1991). Visual curve tracing properties. *Journal of Experimental Psychology: Human Perception and Performance*, *17*, 997–1022.

Kirsh, D. (1990). When is information explicitly represented? In P. Hanson (Ed.), *Information, language, and cognition.* Vancouver, British Columbia: University of Vancouver Press.

Kosslyn, S. (1990). Mental imagery. In D. Osherson, S. Kosslyn, & J. Hollerbach (Eds.), *Visual cognition and action*, pages 73–98. Cambridge, MA: MIT Press.

Lerdahl, F. & Jackendoff, R. (1983). *A generative theory of tonal music.* Cambridge, MA: MIT Press.

Minsky, M. (1986). *The society of mind.* New York: Simon and Schuster.

Neisser, U. (1967). *Cognitive psychology.* New York, NY: Appleton-Century-Crofts.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Newell, A. & Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition.* Hillsdale, NJ: Lawrence Erlbaum.

Norman, D. A. (1988). *The psychology of everyday things.* New York: Basic Books.

Park, D. C. & Shaw, R. J. (1992). Effect of environmental support on implicit and explicit memory in younger and older adults. *Psychology and Aging*, *7*, 632–642.

Raiffa, H. (1968). *Decision analysis: Introductory lectures on choices under uncertainty.* Reading, MA: Addison-Wesley.

Reason, J. (1990). *Human error.* Cambridge, England: Cambridge University Press.

Reeves, A. & Sperling, G. (1986). Attention gating in short-term visual memory. *Psychological Review*, *93*, 180–20.

Sereno, M. I. & Allman, J. M. (1991). Cortical visual areas in mammals. In A. Levinthal (Ed.), *The neural basis of visual function*, pages 160–172. London: Macmillan.

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

R. Simmons, D. Ballard, T. Dean, & J. Firby (Eds.). (1992). *Control of selective perception*, Stanford University. AAAI Spring Symposium Series.

Sperling, G. (1960). The information available in brief visual presentations. *Psychological Monographs*, *74*.

Stigler, G. J. (1961). The economics of information. *Journal of Political Economy*, *69*, 213–285.

Tarjan, R. E. (1985). Amortized computational complexity. *SIAM Journal on Algebraic and Discrete Methods*, *6*, 306–318.

Tarr, M. & Pinker, S. (1989). Mental rotation and orientation-dependence in shape recognition. *Cognitive Psychology*, *21*, 233–282.

Tate, A., Hendler, J., & Drummond, M. (1990). A review of AI planning techniques. In J. Allen, J. Hendler, & A. Tate (Eds.), *Readings in Planning*, pages 26–49. Morgan Kaufman.

Treisman, A. & Souther, J. (1985). Search asymmetry: A diagnostic for preattentive processing of separable features. *Journal of Experimental Psychology: General*, *114*, 285–310.

Ullman, S. (1985). Visual routines. In S. Pinker (Ed.), *Visual cognition*, pages 67–159. Cambridge, MA: MIT Press.

Waltz, D. (1975). Understanding line drawings of scenes with shadows. In P. H. Winston (Ed.), *Psychology of computer vision*. Cambridge, MA: MIT Press.

Author Notes

We thank Steve Haehnichen for his work on the initial implementions of Tetris and RoboTetris. Charles Elkan, Jeff Elman, Nick Flor, Jim Hendler, Ed Hutchins, and Teenie Matlock provided helpful comments and suggestions on early drafts of this paper.