Error correction in large-scale cognitive maps

Ernest Davis

Courant Institute
251 Mercer St., New York, New York 10012

ABSTRACT

A mobile robot that maintains a dynamic cognitive map will often find that the information in the map is contradicted by his perceptions, and is therefore incorrect. Such errors may be the result of an earlier misperception, an erroneous matching, an erroneous default inference, computational errors, a change in the world over time, or an erroneous previous error correction. Due to the complexity of inference in forming cognitive maps, domain-independent strategies for error correction, such as data-dependencies or conditional probabilities, are not sufficient by themselves to give a robust error correction scheme. Rather, domain-specific techniques and heuristics must be applied. We discuss some of the basic issues involved in detecting, diagnosing and correcting errors in the cognitive map. We also discuss how a robot may decide whether to take actions in order to gather relevant information.

1. INTRODUCTION

Most animals, over the course of their lifetime, move about in an environment that is considerably larger than the range of their sensors and is not under their direct control, but is stable enough that many features remain the same between one visit to a place and the next. Such an animal, or a mobile robot with sensors that operates under similar circumstances, stands to gain from learning and remembering the geographic characteristics of its environment. A *dynamic cognitive map* is a knowledge structure that supports such learning and remembering. There have been numerous studies[1-9] proposing knowledge structures for dynamic cognitive maps for use in AI systems.

It is not possible, in general, to design a cognitive mapping system that is both powerful enough to be useful to a robot in a rich environment and also secure enough to be guaranteed correct. In almost any uncontrolled environment, there are possible circumstances in which the cognitive mapping system may perform an operation which, though reasonable, is in fact mistaken and results in the cognitive map being incorrect. A robust cognitive mapping system must therefore have the capacity to detect and deal with errors. Previous cognitive mapping systems that have avoided doing error correction are necessarily fragile. A number of systems[4,7-10] have addressed the problem of correcting perceptual errors, but these have generally used overly simple models that are appropriate only in very restricted environments. In this paper, we give a preliminary discussion of the issues that arise in error correction in a broad range of cognitive mapping systems. We do not, however, give any complete algorithms for any specific cognitive mapping architecture.

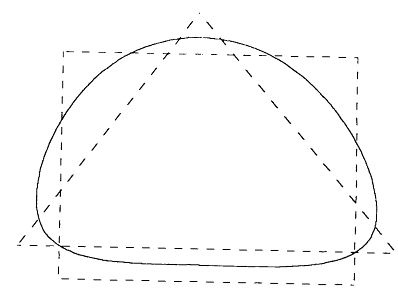
Error correction is an important issue in many areas of AI, and a number of domain-independent techniques have been developed to address it, such as data-dependency maintenance or numerical combination of evidence weights. We shall show below (section 6) that, though these techniques may be helpful in dealing with cognitive maps, they are not by themselves sufficient, owing to the complexity of the inference performed in cognitive mapping. Rather, error correction in cognitive maps requires a domain-specific architecture and heuristics. In other words, you cannot take an existing cognitive mapping system and make it robust by connecting it to a truth-maintenance system, or by throwing in conditional probabilities.

2. FUNDAMENTALS

A cognitive map is a knowledge structure that connects spatial properties to real-world features. Features may include named objects such as "the Statue of Liberty"; object characteristics, such as "a large beech tree"; continually varying measurements, such as "surface reflectance" or "elevation" in a two-dimensional map; or quantified statements such as "an area with no trees" or "an area where every house has a TV antenna". Spatial properties may include exact coordinates; shape characteristics; metric relations between regions, such as distance or direction; topological relations between regions, such as containment or adjacency; or characteristics of spatial distribution, such as "Every fifty to one hundred feet". (These lists are intended to be suggestive rather than exhaustive.)

A cognitive map describes an external environment; it may be correct, incorrect, or partially correct, depending on the actual state of the world. The *meaning* of a particular cognitive map may be defined in terms of the assignment of correctness over the space of possible real-world states. The rules that define this meaning are the *semantics* of the representational system. Since almost all cognitive maps are incomplete or imprecise in some respects, it is worthwhile being systematic about the semantics of a representation, rather than, as is common, letting the semantics be defined operationally by the behavior of the program that manipulate the map. Otherwise, it becomes easy to leave potential sources of ambiguous interpretation unresolved, or to have two modules of the system interpret the same representation in conflicting ways. We illustrate with two examples:

1. Shape: Many cognitive maps approximate the shapes of real-world features, which may be complex or ill-defined, in terms of an idealized simple geometry. In such cases, it is often possible to find two quite different legitimate representations for the same actual shape (Figure 1.)



Either of the dotted shapes is
a reasonable approximation
for the solid shape

Figure 1: Two representations for a single shape

It is therefore important to define the sense of approximation involved, so that sound rules for matching can be found. Consider, for example, the following approximation criteria for two-dimensional shapes:

i. The approximation boundary is everywhere close to the real boundary.

ii. The real boundary is everywhere close to the approximation.

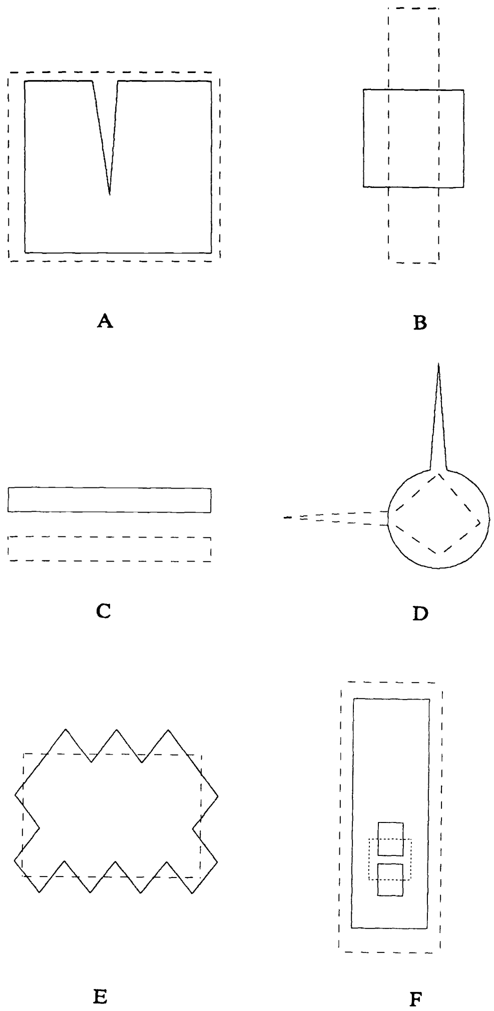
iii. There is a continuous 1-1 function from the approximation interior onto the real interior that moves points only a small amount.

iv. The area of the symmetric difference between the two regions is small.

v. The tangent to the approximation is close to the tangent to the real boundary at some nearby point.

Different approximation criteria lead to different evaluations of correctness. For example, in figure 2, example A satisfies criteria (i), (iv), and (v); B satisfies (ii); C satisfies (i), (ii), (iii), and (v); D satisfies (iv); E satisfies (i), (ii), (iii), and (iv); and F satisfies (i), (ii), (iv), and (v).

2. Individuation of objects: In a cognitive map that enumerates discrete objects, two questions arise about the enumeration: (i) Is it complete? That is, can it be assumed that any object that is in the area shown in the map and that can be detected by the sensors is represented in the map? If



Real shapes in solid.

Approximations in dotted lines.

Figure 2: Criteria of approximation

not, is there any way to express in the map that some particular area is complete with respect to some type of object? Is there ever any way to infer from a map that a object of a particular kind is not present in an area? (ii) Are the objects enumerated all distinct? That is, is it possible for an object to appear twice in the map under two different geometrical description? If only separated pieces of an object have been seen, is it legitimate for them to be represented separately in the map?

What semantics should be chosen depends on the type of information available and the use being made of the information. For example, if the objects being recorded are large and the robot passes close to them, then local properties of objects will be particularly evident, and the semantics should allow them to be represented. Another example: as noted above, if the robot often sees only parts of objects due to occlusion, it is probably unwise to require the map to identify two separated parts of an object as parts of the same object.

The cognitive maps we are considering are built up dynamically from the perceptions of a mobile robot. That is, a robot moves about in an environment, receiving information about its motion from its effectors, and information about the local scene (and possibly about its motion, as well) from its perceptions. The perceptual information may be either in the form of a discrete series of scene descriptions, or a characterization of the continuous evolution of the perceptible scene. Updating the cognitive map involves relating the perceptual information to the cognitive map and to incorporating any new information in the perceptions into the map. Both of these operations become much more difficult if the map's description of the area being perceived is incorrect. Relating perceptions to the map is difficult because perceptions correspond only to the correct information in the map; they contradict the errors in the map. These contradictions will interfere with the task of finding the valid correspondences. Incorporating new information must now deal with correcting information in the map, not merely adding information to it.

When the robot returns to an area previously seen, relating perceptions to the map involves finding an explicit match between its current perceptions to the previous record of the area in the cognitive map.* This match can be *forced,* meaning that the identification holds necessarily if both the map and the perception are correct, or it may be *optional,* if the identification is not a necessary consequence of the map and perception, but is merely suggested by the similarity of the perception to the map. The need to find these kinds of matches suggests that it is not sufficient to index cognitive maps by position; there should also be a feature-based index.

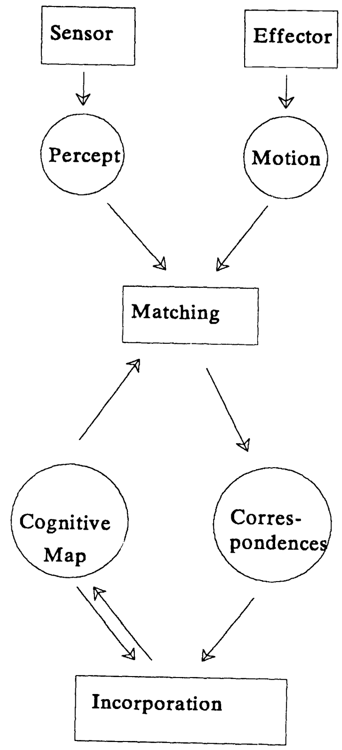


Figure 3: Architecture of a Cognitive Mapping System

3. SOURCES OF ERROR

Errors can enter the cognitive map from a number of different sources:

1. Errors of perception [8–11]. Most image interpretation systems can only give a best guess as to interpretation; by the nature of the task, they cannot give an answer that is guaranteed correct. The perceptual information may contain either errors in geometry or misidentifications of features. Incorporating a margin of error into the representation, by using relational constraints[2,3,9] alleviates the problem but does not eliminate it. Errors tend to follow a probability distribution rather than to have fixed bounds; therefore, any bounds tight enough to give useful information are likely to be exceeded occasionally. These relational constraints are also difficult to compute with.

2. Errors in effector feedback, giving incorrect information about the motion of the robot.

* Matching features in the current perception to the same features in the immediately preceding perceptions is a much easier and more reliable operation. The failure to distinguish between these two types of matching was a major limitation of MERCAOTOR.[3]

3. Matching errors. An optional match is a best guess, rather than a sound deduction, and therefore can be in error, if the robot identifies two distinct but similar places. If the semantics rules out representing a single feature twice in the map, then the failure to find a correct match may also introduce an error into the map.

4. Heuristic error. In many cases, assembling a cognitive map requires addressing problems that, in principle, are computationally intractable in the worst case. In such cases, the system may use approximate or heuristic methods; these will sometimes give incorrect solutions that must be corrected later. For example, the SPAM and MERCATOR programs[2,3] used a combination of hill-climbing and Monte Carlo techniques to extract information from a collection of constraints on distances and direction, a problem which is NP-hard.[12] These techniques are not logically sound and therefore tend to introduce into the map new information that may not be correct.

5. Default inference. The cognitive mapping system may flesh out the map using a variety of default inferences which may go wrong. For example, if you are traveling on a road, and you see a traffic light, you can usually infer that there is a cross-road there, but this inference may be wrong.

6. Temporal change. The world may have changed since it was last seen.

7. Error introduced in error correction. As we shall discuss below, there are usually several different ways of changing a cognitive map to resolve any particular problem. Choosing the wrong correction may introduce new errors.

Once an error is introduced into a map, normal inference process can easily spread it around to generate new errors elsewhere in the map. For example, suppose that you have previously seen that the turn-off from Atlantic Avenue onto Maple Lane is opposite a gas station, and your cognitive map records the fact. If the gas station is now taken down, and you are driving up Atlantic Avenue, you may go past Maple Lane without realizing it, and place all the features you see on the wrong side of the Maple Lane turn-off. (Figure 4) My own experience in MERCATOR[3] was that, once a small error was introduced into the map, it spread itself around the map amazingly quickly; however, MERCATOR may have been unusually fragile in this regard.

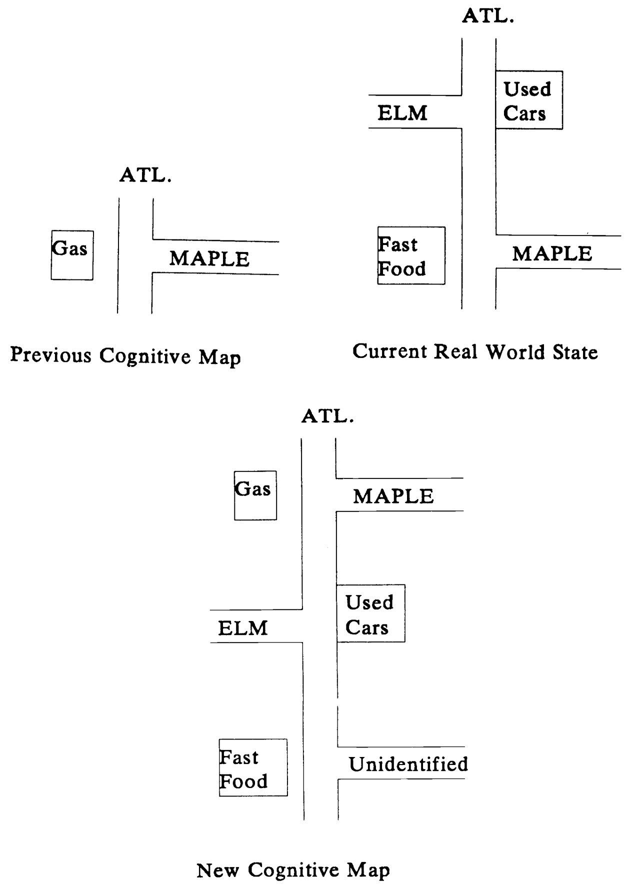


Figure 4: Error propagation.

4. DETECTING ERRORS

In most cases, error is detected when the cognitive mapping system observes that the current perceptual information is inconsistent with the cognitive map. This inconsistency may be logical — e.g. the map asserts that a field is empty, while the perceptual system reports that it contains a tree — or it may involve a violation of some domain rule — e.g. the perceptual system reports seeing a building over a place where the map records a lake, violating the default rule that buildings are not generally built on lakes. Such a contradiction implies or suggests that either the cognitive map or the perception is erroneous; it is the task of the error correction system to choose between these.

There are also other cases where error correction is appropriate, even though there is no explicit contradiction between perception and the cognitive map. One case is where the perceived scene very closely matches a known place which is marked in the cognitive map as being far from the robot's current position. In this case, it may be more probable that an error has been made in judging position than that the world should contain two such similar places. For example, suppose you set out for a walk from your hotel in a strange city. After walking for a while, you judge that you are probably a mile from your hotel. Suddenly, you come across a building which is indistinguishable from your hotel in all respects. In this case, it is probably more reasonable to suppose that you have made a mistake evaluating your position than to suppose that there are two such similar buildings in the city, though the latter supposition does not violate any actual rules. Similarly, if the perceptions match an area in the cognitive map except for some minor differences, it may be more reasonable to suppose that the differences are due to some kind of error than that such a coincidence of common features could occur.

It is often computationally infeasible to perform a complete check of whether the perception is consistent with the map. In such cases, it is possible for contradictory information to be merged directly from the perception into the map, leaving the map in an inconsistent state. Such contradictions may emerge as the result of some later computation, that brings the two facts together in a way that makes their contradiction evident. If so, the error should be corrected at that time by purely internal examination of the map, independent of the current perception. Similarly, implausible duplications of features may lay hidden in the map, to emerge and be corrected at a later date.

5. MATCHING

The interactions between matching and error correction are particularly difficult. Matching errors can be especially malignant in their effects. A match between two similar regions that are far apart can introduce very substantial distortions into the map, for all the environs of each region will similarly be placed close together. If these environs are sufficiently similar to one another to be yield plausible partial matches, the map can get thoroughly twisted up with combinations of facts that belong to two different parts of the world. Unraveling such an error involves separating information from the two sources. It would seem advisable to maintain two separate tokens for the two perceptions of the object with an explicit equality link between them. Over time, however, this becomes burdensome; it does not seem reasonable to maintain a separate token representing each time an object has been seen.

These considerations suggest that it may be well to be very conservative in matching; to delay committing the map to a match between two areas until so many common features have been detected that the probability of coincidence is absolutely negligible. This should apply even to forced matches, to avoid spurious matches generated by previous errors.

6. CREDIT ASSIGNMENT

The problem of determining what basic error underlies an observed failure, known in learning theory as the *credit assignment* problem, is particularly difficult in dynamic cognitive mapping, because natural chains of inference can extend over long periods of motion and perception, making basic calculations dependent on virtually all previous measurements and judgements. Consider, for example, a mobile robot that tracks its own position using a combination of dead reckoning together with perceptual feedback, and which uses this judgement of position in its identification of perceived objects. If this robot mismeasures its motion in an early stage of its travels, then this can throw off all subsequent judgements of position. Moreover, if the robot also uses its perception to judge its position, then any type of error — misperception of geometry, misidentification of features, mismatch, or the like — can propagate in the same way. This kind of propagation can lead to seriously distorted maps, as in the example in section 2 of the gas station on Maple Street. More subtly, but perhaps even more seriously, it means that when a contradiction is detected, the cognitive mapping system must consider all possible previous sources of error, which means virtually all previous perceptual information and plausible inferences. Moreover, if the robot wants now to compute the effects of changing its judgments of three hours ago, it may require recomputing every change it has made to its map since then, as all these changes were based on a different judgement of position. Considering *all* possibly relevant changes to the map to determine which correction best fixes the observed problem would thus be a computational nightmare. Since the dependencies here are genuine, they are not avoided by any general scheme of belief revision, such as data-dependency maintenance or the use of conditional probabilities. Algebraic techniques for tracing from a failure to a underlying error, such as those used by Simmons,[13] may provide some degree of pruning. However, they are unlikely to constitute a complete solution, due to the complex interrelations of data used in computing an answer.

The problem will be compounded if, as is common, the input perceptual information is not incorporated directly into the map, but, rather, is translated to some more tractable or compact form. If the original information is thrown away in the process, then access to the original source of error becomes impossible. Even if the original input is maintained as a justification for the compact form, determining the effect of making a correction to the original input will require recomputing all the transformations.

These considerations suggest some possible approaches to designing systems with error correction. The first would be to prevent the system from constructing very long chains of inference, such as that involved in tracking absolute position in the scenario above; so to speak, to erect fire-walls against very long searches through the space of underlying assumptions. Such a strategy would probably involve creating a very localized map, with little explicit global information. Where a global inference must be made, it should be a very robust one, which is valid under a wide range of possible errors. Note that any error in orientation can generate an error in position proportional to the total distance travelled since the error was made.

Another possible approach would be to give up on trying to find an original cause for the error, and simply look for a plausible new state of the map. This would be particularly appropriate, of course, in maps where the original form of the input has been discarded. For example, if the robot is found to be somewhere else than expected, don't try to figure out where the process went wrong, just adjust the map to record his new position and his recent path. Such a strategy will often result in many errors persisting in the map past the time when they could, in principle, have been eliminated, and, quite possibly, in the map containing global inconsistencies; but it may be the most effective solution computationally.

7. HEURISTICS FOR ERROR CORRECTION

In general, there will be many possible ways of changing the map or the perceptual information to resolve any given conflict. It is the task of the error correction mechanism to find the "best" such correction.

The first condition on correction is that the final map make sense internally. It must be logically consistent (e.g. the map should not contain one datum that area A is empty, and another that object O is in A), geometrically consistent, and consistent with physical constraints (e.g. it should not record that two incompatible features are in the same place.) In many cases, a complete verification of these constraints may be computationally infeasible; if so, some partial conditions will have to suffice.

In many structures for cognitive maps, the data in the map are interrelated by these constraints, so that, if one datum is changed, it will be necessary to change others as well, in order to keep the map consistent. For instance, if the map records each of the lengths and angles of a triangle, and the error corrector decides that one of these quantities is in error, it will have to change at least two of the other quantities as well, since there are only three free parameters in a triangle. There are structures for maps in which each datum is independent, so that the map is always consistent no matter what combinations of values are specified. An occupancy array has this property; the occupancy of any square of an array does not depend on the occupancy of any other square. Another structure with this property is a map in which objects are organized in a free tree, and the map records the relative position only of adjacent objects.[4] Even in these maps, however, it will often be more reasonable to change many data rather than to change a single datum. Consider, for example, a map using an occupancy array that records that two objects are adjacent. If it is later found that position of one of these objects is misrecorded in the map, and that it is, in fact, ten feet away, then it is often reasonable to "move" the second object along with the first. That is, we would assume that the implicit information relating the rooms to the walls is more reliable than the explicit record of the position of the rooms. Therefore, in formulating an error correction strategy and deciding which information to preserve and which to abandon, it is advisable to consider as potentially relevant all information implicit in the map, not just that explicitly expressed. Error correction strategies that ignore these implicit constraints, such as [8], necessarily apply to a limited range of environments.

In some cases, it is reasonable to organize the map so that it directly employs the most reliable information. In the above example, it would be feasible to have the adjacency of the two objects directly represented; in fact, the accessibility of this fact would simplify many important inferences as well as aiding in error correction. In other cases, however, particularly reliable information may be quite unhelpful in most uses of the cognitive map. For example, if the robot sees two objects simultaneously, then the datum that these objects can both be seen from a single position is as reliable as the identification of the objects; substantially more reliable than data such as the distance between them or their relative orientation. However, since this datum involves not only the position of the two objects but also the absence of other occluding objects, it is very difficult to use in a forward direction as a basis for other inferences, and therefore should not be part of the basic structure of the map. The best that can be done with it is to record it as a constraint, and then use it prune the space of positions of objects, in a "generate and test" manner. (Note that, though this fact depends on the positions of all other objects, it may be more reliable than the judgement of positions of other objects; however, it is not more reliable than the temporal inference that no other objects have moved to interrupt the view.)

Finding the most likely correction involves evaluating the reliability of each relevant datum in the map, and the likelihood that it is in error. This evaluation depends strongly on the source of the information:

Direct perception: The error corrector should have available information about the reliability of various kinds of data acquired through the sensors. For example, in humans, it would seem that object type identification is fairly reliable, while metric judgements is relatively uncertain. The reverse would probably be true in a robot with range-finders and current software for object recognition. Similarly, the error corrector needs information about the reliability of effector feedback in reporting the motions of the robot. This information about the reliability of perception is also used to determine the likelihood that it is the current perception rather than the existing map that is in error.

Inference: If a datum ϕ is inferred from data $\theta_1, \theta_2 \cdots$ then the likelihood of ϕ is some function of the likelihoods of $\theta_1, \theta_2 \cdots$ — their product, if they are independent probabilities. If the inference involves the use of an uncertain heuristic, then that uncertainty must be factored in as well. Hence, default rules used by the system should be tagged with a measure of their reliability. Similarly, metric quantities that are calculated using numerical techniques of limited accuracy should be tagged with some additional measure of uncertainty.

Temporal change: The likelihood that a given state continues to hold after it is perceived tends to decrease over time. The rate of this decrease depends on the type of state involved. If one returns to a city street after fifteen minutes absence, one expects to find many of the parked cars in the same place, but not the moving cars; after a week, one expects to find that many of the cars have moved, but the buildings are the same; after ten years, that some, but not necessarily all of the buildings are the same; after five hundred years, that the general topography is the same. In the absence of information about relevant events that change the state, the likelihood that no change has occurred may be taken to be a decaying exponential.[14] It is therefore necessary to tag each datum in the map with an indication of the time when it was last perceived.

Corrections due to temporal change often changes collections of data in quite different ways from other types of correction. For example, suppose that you believe that your radio is in the trunk of your car which is parked legally on Broadway and 44th Street. Now you come back to Broadway and 44th Street, and you do not see your car. If you were mistaken about where you parked your car, then it is likely that your radio is still in the trunk; however, if your car has been moved (presumably by unauthorized persons), then it is likely that your radio is no longer in the trunk. Similarly, as noted above, the reliability of the perception that a view of an object is unoccluded is unaffected by errors in judging the positions of potentially occluding objects, whereas it is affected by the possibility that these objects have moved into occluding position. By contrast, suppose that you believe you have left your driver's license on your bureau and you come back and it is not there. If the error is due to error, then the license is lost, and there is no particular reason to believe that it is anywhere in the house. If the error is due to temporal change, then presumably your spouse put it away somewhere, and it probably is in the house. The point is that a temporal change often implies or suggests some event causing the change, and this inferred event may in turn imply further changes or constancies.

Repeated Evidence: A datum gains credibility if it is confirmed by many sources. If all the sources are independent, then the probability that the datum is erroneous is the product of the probabilities of error in each individual source. The problem is that determining that sources are independent or evaluating the degree of dependence can be tricky. Two sources may be independent as regards one kind of error but not as regards another. For example, if the type of an object is identified twice based on two observations at different times from the same viewing points, then these identifications may be independent pieces of evidence as regards certain potential disruptions of the perception, such as noise, but not as regards more systematic distortions, such as occlusions or partial views. Observations at different times have little effect in evaluating the likelihood of temporal change; only the latest observation matters.

Optional matches: Optional matches introduce errors if the two areas being matched are merely similar and not identical. Judging whether the match should be withdrawn requires evaluating the likelihood of such an extensive coincidence between features. This likelihood should be estimated and recorded when the match is originally performed. In the case of a missed match, it is obviously not generally feasible to evaluate the probability of error at the time the error is made; this evaluation must be made when the error correction proposes this match as a solution to its problems.

Error correction: The likelihood of facts introduced by the error correction must be computed from the previous likelihoods, conditioned on the fact that the detected problem has, in fact, occurred. Furthermore, the likelihoods of all other data in the map that could be relevant to the detected problem should likewise be updated. For example, suppose that the map initially records two measurements ϕ_1 and ϕ_2 with reliabilities p_1 and p_2, where $p_1 < p_2$. Now the system detects a problem that must be due to one or the other of these. Since ϕ_1 is the less reliable, the system decides to replace it by the new measurement ϕ_1'. The reliability of the new measurement ϕ_1' depends on the likelihood that the error corrector made the correct choice. Therefore, it is an increasing function of p_2 and a decreasing function of p_1. Moreover, the reliability of ϕ_2 should be decreased, since the discovery of the problem raises the likelihood that ϕ_2 is in error. If some later perception forces ϕ_2 to be changed then ϕ_1 will have to be reconsidered. Bayesian updating gives a principled way of changing probabilities that ordinarily has all these properties. It requires, however, either extensive information about conditional probabilities or extensive use of independence assumptions. In complex situations, the former may be unavailable, while the latter may be unjustified. If so, some more *ad hoc* techniques must be employed.

8. ACTIVE PERCEPTION

One possible response to detecting an error in the cognitive map is to go out actively and look for disambiguating information. (This an instance of the use of "active vision" to obtain a particular desired piece of information.[15,16]) The actual decision to do this obviously involves many issues external to the cognitive mapping system; the robot must decide whether he can afford the time and resources based on the state of his other tasks and the resources available. However, information provided by the cognitive map is central to determining what kind of information to look for and where to look for it. If the error corrector considers it reasonably likely that the problem discovered is a result of an error in current perception, the robot can simply look again, perhaps with some slight motion to correct for features such as non-general position and highlights. If the system suspects that an error lies in some feature of the map close to the current position of the robot, it can go and check it out. If the system suspects that a temporal change has occurred, the robot can look around for other consequences of the event that caused the change. If the system finds two possible candidate areas in the map both of which match current perceptions, it can calculate some feature that serves to discriminate the two areas, and then go verify that feature.

A number of special heuristics apply when the robot is lost: that is, when he is unable to determine the relation between his immediate surroundings and the remainder of the cognitive map. The robot may physically backtrack; that is, to reverse the steps that got him to this place, as far as he can. It may head for a place, such as a high point, where he will be able to get a good view of a large area, and thus have a chance of seeing some known landmark. It may search in some systematic fashion, such as a maze-searching strategy. If it has some general idea of its position and orientation, it may head off in the general direction of some large or conspicuous object, such as a river or tower, which it will be able to see and recognize anywhere in a large vicinity. It may use special features of the type of the region; for example, if it is looking for a river-bed in a valley, it is usually a good rule to head downward. To generate and verify these kinds of plans, the robot needs a theory relating its perceptual powers to the surrounding environment.[17]

9. CONCLUSIONS

We have seen that error correction is not a feature that can be added to a cognitive mapping system as an afterthought, but, rather, the need for robust error correction must be taken into account in all stages of designing the cognitive mapping system. In particular:

- The map should be structured so that inferences can be based very largely on local inference; inferences requiring the compilation of many local measurement to give a global constraint should be avoided.
- Data in the map should be tagged with some measure of reliability and with a time-stamp.
- The matcher should be exceptionally conservative in its actions, and should not commit itself to a match until there is so strong a correlation between features to be beyond coincidence. At the same time, it must be forgiving of a certain number of discrepancies between perception and the cognitive map, due to error or change
- In an time-varying environment, the mapping process should have access to a theory of plausible change, so that it can know what changes to expect to come together.

Designing such a system presents many difficulties for which we do not currently have a solution. Research for the near future should focus on developing systems for very restricted environments, either natural or simulated.

10. ACKNOWLEDGEMENTS

This research was supported by NSF Grant #IRI-880529.

11. REFERENCES

1. B. Kuipers, "Modelling spatial knowledge," *Cognitive Science*, Vol. 2, No. 2, 1978, pp. 129-154.

2. D.V. McDermott and E. Davis, "Planning routes through uncertain territories," *Artificial Intelligence*, vol. 22, 1984, pp. 107-156.

3. E. Davis, *Representing and Acquiring Geographic Knowledge*, Pitman Publishing, London, 1986.

4. R. Smith and P. Cheeseman, "On the Representation and Estimation of Spatial Uncertainty," Tech. Rep., SRI International Robotic Lab, 1985.

5. W. K. Yeap, "Toward a Computational Theory of Cognitive Maps," *Artificial Intelligence*, vol. 24, no. 3, 1988, pp. 297-360.

6. B. Kuipers and T. Levitt, "Navigation and Mapping in Large-Scale Space," *AI Magazine*, vol. 9, no. 2, 1988, pp. 25-46.

7. T. Dean, "On the Complexity of Integrating Spatial Measurements," this conference.

8. H. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," *AI Magazine*, vol. 9, no. 2, 1988, pp. 61-74.

9. R. Brooks, "Visual Map Making for a Mobile Robot," in M. Fischler and O. Firschein (eds.) *Readings in Computer Vision*, Morgan Kaufmann, 1987.

10. M. Beckerman, J.P. Jones, R.C. Mann, and S.E. Johnston, "Spatial reasoning in the presence of systematic sensor errors", this conference.

11. R. Bajcsy, E. Krotkov, and M. Mintz, "Models of Errors and Mistakes in Machine Perception," Tech. Report MS-CIS-86-26, GRASP Lab 64, U. Penn, 1986.

12. Y. Yemini, "Some theoretical aspects of position-location problems," *Proc. 20th Symp. on the Foundations of Computer Science*, 1979.

13. R. Simmons, "A Theory of Debugging Plans and Interpretations," *Proc. AAAI-88*, pp. 94-105.

14. T. Dean and K. Kanazawa, "Probabilistic Temporal Reasoning", *Proc. AAAI-88*, pp. 524-528.

15. T. Garvey, "Perceptual Strategies for Purposive Vision," SRI AI Center, Tech. Report 117, 1986.

16. P. Selfridge, "Reasoning about Success and Failure in Aerial Image Understanding," Tech. Report 103, U. Rochester, 1982.

17. E. Davis, "Inferring ignorance from the locality of visual perception," *Proc. AAAI-88* pp. 786-791.