Design Of a Highly Parallel Visual Recognition System

Daniel Sabbah

University Of Rochester
Rochester, New York 14627

ABSTRACT

Relaxation in a conceptual hierarchy is proposed as a useful principle in the design of a visual recognition system. The domain of the implementation is Kanade's Origami world. The hierarchy is described along with the connection patterns that define the complete network. Advantages of this system are a uniform approach to intermediate and low level vision, inherent parallel approach and sharing of partial results to dynamically cut down the search for a correct match.

1. Introduction

The paper argues for relaxation in a conceptual hierarchy as a useful principle in structuring parallel formulations of visual recognition tasks. A conceptual hierarchy is defined as an active semantic network of computing *units* logically partitioned into abstration *levels.* Each level is in turn defined by unique sets of parameters enumerated over a fixed grain. These parameter spaces form the basis for recognizing the features associated with the various levels. Relations extracted from physical constraints imposed by the world domain produce the connection patterns necessary for communication among the units in the network. Relational connection patterns represent an extension to strict next neighbor communication of previous relaxation networks used in visual pattern recognition. Goal directed vision uniformly blends into the overall recognition computation and provides model directed disambiguation at all abstraction levels. lateral Inhibition also plays a role in dealing with various forms of missing information and (gaussian) noise in the input.

Previous work has successfully concentrated on parallel computation of low level image processing functions [DavRosen, Ballard, Zucker]. Rosenfeld's methods of cooperative "fuzzy" (probabilistic) computation use relaxation as a control mechanism and next neighbor relations as connection patterns [Rosen21. Although these methods work well, even on noisy data, they do not readily extend to the multiple abstraction level approach necessary to attack higher level recognition tasks. Ballard's hough transformsfBallardl do allow generalization to multiple levels necessary tor recognition of complex visual features but do not allow for dynamic interaction of partial results and feedback that a control mechanism such as relaxation permits. Evidence for relaxation as a control paradigm in higher levels of vision is pointed out by Hinton. His puppet finders use relaxation of multiple hypotheses about existence of puppets in a scene to do segmentation and labeling[Hinton]. Unfortunately, his system says little about the role of low level vision. The method presented here uniformly manages computation of both low level intrinsic functions(pre segmentation tasks) and intermediate level feature extraction (segmenting the input into semanticaly meaningful units).

This work was supported in part by DARPA under Contracts N00014-78-C0164 and N00014-80-C0197. Also by IBM under Graduate Residency Scholarship.

2. Vision World Domain

To build a system incorporating these principles, 1 chose a relatively simple visible world. Kanade's Origami World[Kanadel] provides a domain in which the computations necessary to "see" were straightforward and readily available [Kanade?.]. Origami world was already divided into suitable feature sets and the use of the skew symmetry heuristic made extraction of three dimensional orientation simpler than other known methods. Objects in the Origami world, such as the ones in Figure 1, were easy to represent and yet captured all important aspects *of* building a parallel network.

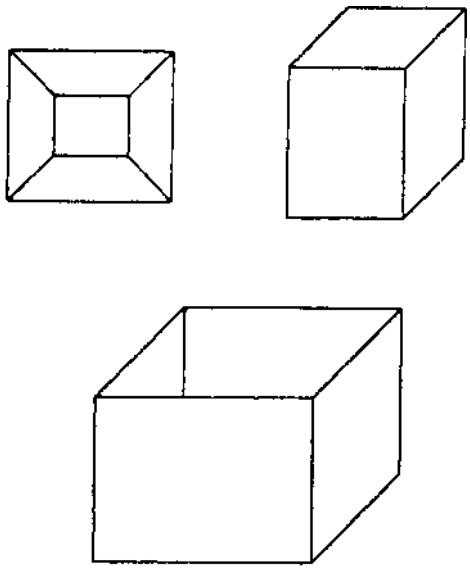


Figure 1. Origami world objects

Kanade's world also proves important in illustrating one of the main advantages of the present formulation. Missing information in the input scene due to self-occlusion forces Kanade to use a parallel line heuristic to disambiguate an otherwise underdetermined problem. He cannot sequentially enumerate all the possibilities in the search space of gradients matched to the search space of allowable objects. This search problem is never encountered by this system. Goal directed disambiguation combined with continual communication of partial results dynamically prunes the search tree thereby eliminating the problem without incorporation of any heuristic.

In terms previously described, objects in this system are complex features (ie many parameters) that have explicit names associated with them such as box, cube, squat, rhomboid prism. Only objects modeled at the various levels of the system are recognized. The purpose here is to label already modeled entities(goal oriented pattern matching). There is no explicit validation of arbitrary Origami world objects by doing "Waltz World" validation labeling. Recogniton of objects can be extended to be view independent and, in fact, allows for the generation of the inverse transformation between instance and model.

Input to the system is in the form of raw edge data, such as that extracted by convolving digitized images to produce primal sketches[Marr]. We assume two dimensional orthographic projection of Origami world objects. The current implementation takes directed line segments as input. Real world input may be added later.

Although Origami world might seem too simplistic, we were able to generalize the representations used in this model to describe more complex 3D rigid solids[BallSabb] and effectively generalize the applicability of this approach to a more complex domain. The general solution assumes appropriate breakdown of the domain into levels (hierarchical component parts) and appropriate computation of suitable intrinsic parameters [Ballard2]. How this was done in Origami world will be the next topic

3. Hierarchical Levels in the Network

Segmenting knowledge about the domain intc abstraction levels is one of the key parts of the recognition algorithm. First, a *level* in this model is defined by:

1. - A unique set of parameters representing the features extracted on that level of abstraction. In effect, this defines the semantics associated with the level although it is the connection pattern that truly implements meaning.

2. - A fixed grain for each of the parameters. For example, the spatial location corresponding to points on the field of view(retinotropic map) is represented by a 2D parameter space: (x,y) with a [10,10] grain. This allows association of any feature with any projected spatial location.

Figure 2 shows the features associated with the various abstraction levels of the system.

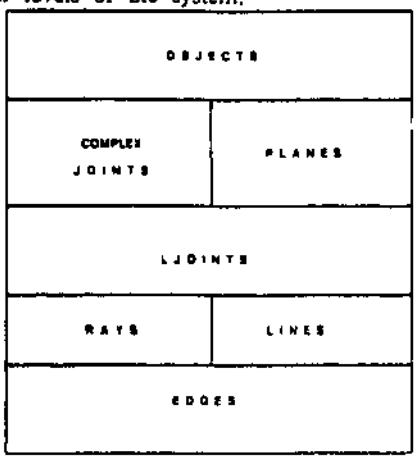


Figure 2. Abstraction levels in the Origami World

Note how the complexity of the features rises as one gets further away from raw image data. This is reflected as an increase in the number of parameters necessary to fully describe the features. Also note that features of level N are always composed of features from level N-1. It is possible to represent independent feature sets on the same level(e.g. the level containing planes and complex joints) implying that both features are composed of elements from the level below.

Examples of the parameters associated with the various levels and associated explanations are shown in table 1.

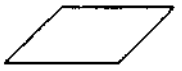
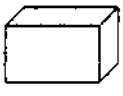
Selected Features	Associated parameters
L-joints 	(x,y) - represents the position in the retinotropic map θ_{ij} - represents the orientation of one of the rays relative to a fixed origin and an axis(arbitrary but consistent choices) φ_{ij} - represents the size of the opening in degrees.
2D SHAPES(skewed rectangles) 	(m,n) - represents the location of an arbitrarily chosen reference point for the shape (comperable to center of mass for the object). $(\Delta x,\Delta y)$ - represents the shape of the object. A ratio relative to a model whose sides are of length 1. (α,β) - represents the measure of skew of the projected object. Results in two ideal points in gradient space that correspond to orientations in space. (S) - represents scale of the object relative to a fixed model value of 1. (θ_ρ) - represents the orientation of the skewed rectangle within the plane of orientation.
3D Objects 	$((\lambda_1,\nu_1),(\lambda_2,\nu_2),(\lambda_3,\nu_3)...)$ - represent the points that describe the orientations of the planes that make up the object in terms of (p,q) gradient space. Currently , this is dependent on the number of visible planes in the object. This assumpuon would have to be eliminated to extend the representation to arbitrary viewpoints [BallSabb]. A number of other parameters exist to represent shape, position and scale to allow complete description of the object relative to an internal model. We discuss the complete representation in[BallSabb].

Table 1. Parameter spaces and associated features

Unique combinations of parameters on a specific level have an active computation *unit* associated with them in the implementation. The unit represents a hypothesis that the feature represented by the unique combination of parameters exists in the input image. The "strength" of the

output of the unit reflects its current confidence level.

As the level of complexity rises, only the "important" combinations of parameters are represented. "Important" means important to recognize in the particular world the system is trying to see. This reduces the number of units drastically since many arbitrarily detailed units can be effectively ignored. Such an unknown combination may be parsed by the system but no unit exists to represent it at the top level. For example, The object in figure 3, a folded paper W, can be parsed up to the plane level in the model, but has no explicit representation in the objects level and therefore causes no explicit recongnition (ie. a single unit to saturate) as such at the top level. The resulting selective enumeration can be seen as the dividing line between low level image parsing and higher level image interpretation. Adding the W-object requires the addition of an unit at the object level and the addition of the connections to that unit. How the unit is connected into the existing pattern is described in the next section.

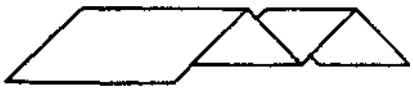


Figure 3. W-folded paper object

4. Connections in the Network

The relations that define the connections of a unit on any level define the semantics associated with it. The units comprising each level behave uniformly regardless of the parameters associated. At present, the only difference between instantiations of units is the connection patterns. Behavior is therefore completely determined by connected inputs to that unit. The connection pattern for any one unit on level N with a group of associated parameters [I..n] is defined by three relations:

1. R_{bu} (bottom up) is the projection of the complete relation resulting in the set of connections mapping units onto the level above. The relation maps one onto many. It is only necessary that the relations allow parameters to be enumerated in some regular way. In most cases the relation can be derived from physical constraints. For example, in the current implementation, all relations are a result of geometric principles, making it possible to generate them in a regular and exhaustive manner.

2. R_{td} (top down) is the projection of the complete relation resulting in the set of connections mapping a given unit onto the level below. The relation also maps one onto many. It intuitively represents the top down, goal directed information feeding back onto the lower levels.

3. R_{jnh} (inhibit) defines the set of inhibitory connections between competing units. For a given unit, elements related in this way exist on the same conceptual level and represent certain predefined ambiguities arising from noise and occlusion. These ambiguities give rise to subsets of probable interpretations for imperfect input. Competition among these alternatives, to be resolved by additional input from bottom up and top down relations, is the method chosen here to deal with this imperfect information.

Each of these three sets is uniquely generated for every unit on every level and corresponds to the static knowledge pool(ie the semantic net) defining the domain. An example of connection sets is given below.

4.1 Connections between ljoints and skewed rectangles

First, to generate bottom up relations, all possible skewed rectangles are hallucinated from a given ljoint. In this case, the skew is a heuristic that allows us to extract possible planes that the rectangle lies in in 3-space. So a given ljoint can be part of some set of skewed rectangles on the plane level where we can generate (α_p,β_p) and σ_p from σ_{lj} and φ_{lj} in a straighforward way since a skewed rectangle is represented by a parallelogram. In the absence of any other constraints, all possible and legal shapes of the rectangle class are considered at all scale parms giving rise to all the planes that the ljoint could (imply) be considered "partof". This is shown in figure 4.

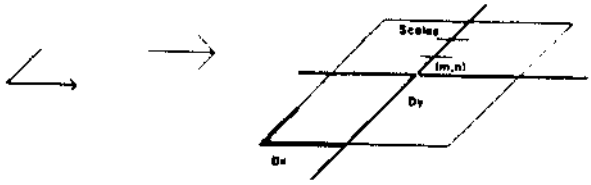


Figure 4. Possible up connections for ljoint to plane.

For the top down connection sets, the inverse relation is used. All possible ljoints are mapped that could have generated the instance of the skewed rectangle represented by the parallelogram. If the description of the plane level parameters is complete (some parameters may be considered spurious to the recognition algorithm) this set is one parallelogram generating only its directly constituent ljoints providing strong feedback indications for the existense of those ljoints in the input scene. Figure 5 shows the resulting ljoints.

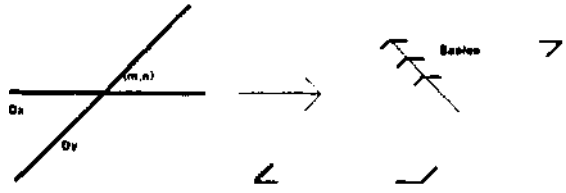


Figure 5, Possible down connections for plane to ljoint.

Paramater values for computation of specific ljoint parms are gotten from simple trigonometry applied to plane parameters. If it had been determined, for example, that scale parms where unimportant to the unique determination of objects in this case, the number of top down connections for this parallelogram would have to be

multiplied by the grain of the scale space since all possible ljoints giving rise to this plane have to be enhanced. Otherwise, possibly valid ljoints receive no feedback.

The third and final example is of the inhibitory set of an ljoint. This portion of the inhibitory set is meant to deal with noisy input (gaussian noise). Figure 6 shows a computation of a next neighbor relation along each of the parameters describing the ljoint and in this case, the inhibitory set. Competing units will serve to sharpen the perception of a specific ljoint at the cost of its logical close neighbors in feature space.

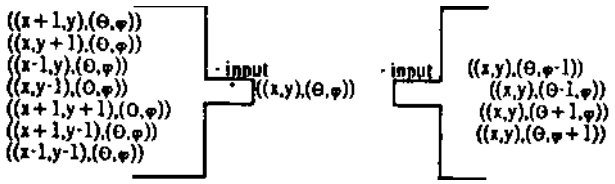


Figure 6. Inhibitory set for ljoints

A value difference of one was used to define next neighbor in this case. One is not a magic value and may be changed depending on the desired immediate scope of influence of the inhibition exerted. A milder form of enlarging the scope of inhibition is realized by letting the effects ripple to larger neighborhoods in later iterations of the overall computation.

A detailed discussion of the connection patterns involved in dealing with occlusion especially on the higher conceptual levels, is too involved for presentation here. Such a discussion is available in [Sabbah],

The overall result for computation of the activation level of a unit is shown in figure 7.

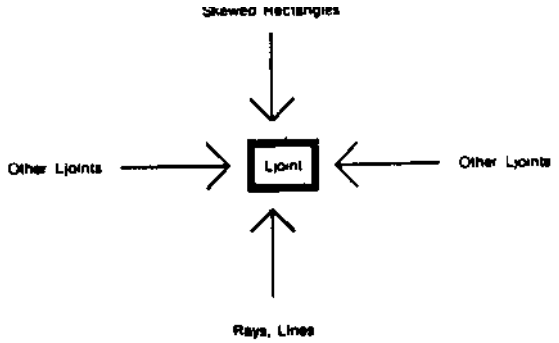


Figure 7. The Unit as a Black Box.

Having detailed examples of the connection patterns among the levels, we now turn to to a discussion of the Black Box of figure 7, the computing unit.

5. Computing with Connections

This section is organized in two parts:

The first discusses the computing unit. The second defines the notion of iterative convergence in this model.

5.1 Computing units

To understand the behavior of an individual unit, activation level must be defined. Activation level is the value that the unit computes at every iteration of the relaxation process. It reflects the current confidence level that the feature represented exists. Since the simulation cannot instantaneously digest all the input necessary to make the recognition decisions, the output of a unit must be a partial result and must be stored as the current level and updated according to the iteration number and the input. The activation level is represented by a real number in the somewhat arbitrary, small range [0,10]. A higher activation level implies a more active and confident state for this unit. It attains maximum confidence at 10 and can go no lower than 0. The max value represents the absolute decision point that the feature really exists in the input image. Enhancement and inhibition across all parms and units proceeds at the maximum rate at this point.

The output of the unit is a direct function of the current activation level and therefore communicates the partial result desired at the current iteration in the relaxation process.

Activation levels are updated according to the following formulas:

$$AL_{N+1} = AL_N + [\text{enhancement input}/\ \text{maxnumber of enhance input}] - [\text{Inhibitory input}/\ \text{maxnumber of inhibition input}] - [K\ (\text{delta t})^2]$$

$$AL_{curr} = \max(0,\ AL_{N+1}) \text{ if } AL < \text{max activation value}$$
$$AL_{curr} = \min(10,\ AL_{N+1}) \text{ if } AL >= 0$$

where the first two terms are straightforward, namely normed input values added to the current stored activation value. When (delta t) is equal to one, the last term represents a measure of self inhibition. Self Inhibition is included to make initial response sluggish. Without it the behavior exhibited is strictly linear and all possible initial guesses eventually saturate at the maximum activation value. Tremendous amounts of "hallucination" results. When (delta t) is greater than one, the last term becomes a measure of exponential decay exhibited by the unit in the absence of any input. Decay is included to allow incorrect initial "guesses" to eventually die out.

I am currently simulating the model and can as of yet only make simple claims about complex behavior. A scheme for dealing with occlusion and multiple objects is in the process of implementation. Also, studies of small networks with the same properties are encouraging [FeldBall]. Convergence in the system is defined, however.

5.2 Iterative Convergence

Absolute convergence occurs on a unit level when that unit attains the maximum activation level and its inhibitive

neighborhood attains minimum activation. Convergence can therefore occur for multiple units on a single level. Convergence also occurs for units on all levels. This is shown graphically in figure 8.

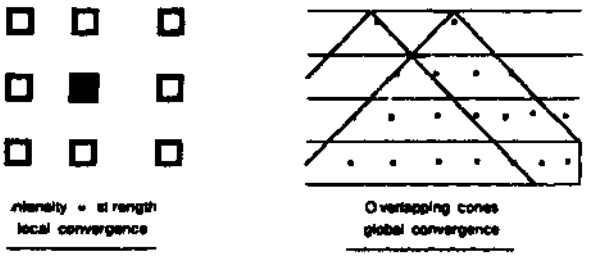


Figure 8. Convergence in the network.

For example, if we are provided with perfect information about a single object in an input scene we get a convergence pattern as in figure 8. In this case we see a cone of convergence . In the case of multiple objects in the scene there may be more than one cone, and they may overlap in their sphere. As long as no two conflicting features (groups of parameters) converge, there is no inherent contradiction in what is seen. Conservation of this property is one motivation for inhibitory connections in the network.

It is not always desirable to converge to a specific single value at the top level of the cone of convergence. If the input information is too noisy or ambiguous (due to excessive occlusion) , the system should rightfully never attain a stable state. If the object in the image is not represented explicitly in network, it should be parsed to the best ability of the lower levels and no convergence should occur at the top of the cone. A cone with the top cut off is the result. It remains to prove that the system converges in "ill behaved" situations. This can only be defined in relation to human performance on the same recognition tasks and assumes the system is used in robot vision.

6. Conclusions

In conclusion, we can see that relaxation in a conceptual hierarchy is a useful principle in visual relaxation tasks for several reasons:

1. It provides a method for uniformly managing the computation of low level intrinsic functions and intermediate level feature extraction. The goal driven segmentation and interpretation of parts of the scene directly influences computation of such low level features as edges and lines. A consistent, understandable role for top down influence blends with bottom up parsing and competition among alternate hypotheses in the presence of imperfect input.

2. Also clear is that in computing the same information as Kanade [Kanade2] the system does not suffer the problems of a sequential formulation. Namely, the sequential formulation of the recognition algorithm must explicitly choose an order for exploring the large search space of alternate hypotheses. The order may or may not result in finding the most suitable hypothesis for the current recognition problem. Exhaustively exploring all possible search strategies is at best time consuming and at worst, impossible. The parallel formulation presented here does not suffer this problem since it allows the pruning of the search tree dynamically by incorporating results from partial computations in the well formed hierarchy. As pointed out by Kornfeld [Korn], this benefit is reaped regardless of whether the algorithm is simulated or actually run on parallel hardware.

3. The system is inherently parallel and therefore can be readily extended to form the basis for parallel architectures.

Finally, we are studying relaxation in conceptual hierarchies as a general paradigm [Sabbah, FeldBall]. it seems to extend to higher levels of abstraction readily. It promises to have nice noise resilience and disambiguation properties. Amazingly enough, it seems to share many similarities with animal vision.

References

[Ballard] Ballard, D.H. (1979), "Generalizing the Hough Transform to detect arbitrary shapes", TR55. University Of Rochester, Rochester, N.Y.

[Ballard2] Ballard, D.H. (1981), "Parameter Nets: A theory of low level vision", TR75, University Of Rochester, Rochester, N.Y.

[BallSabb] Ballard, D.H., Sabbah, D. (1981), "On shapes", accepted for publication in proceedings of IJCAI 1981

[DavRosen] Davis, L.S.,Rosenfeld. A., (1980),"Cooperating Processes For Low Level Vision: A survey", TR 851. University of Maryland, College Park, Maryland.

[FeldBall] Feldman, J.A.,Ballard, D.H., (1981), "Computing with Connections", TR72. University Of Rochester, Rochester, N.Y.

[Hinton] Hinton, G. (1977) /The Role of Relaxation in Vision", Ph.D. Thesis, University of Edinburgh, Edinburgh, Scotland

[Kanadel] Kanade, T. (1978),"A Theory of Origami World", TR CMU-CS-78-144, Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pa.

[Kanade2] Kanade, T. (1979),"Recovery of Three Dimensional Shape of an Object from a Single View", TR CMUCS-79-153, Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, Pa.

[Korn] Kornfeld, W., (1979),"Combinatorially Implosive Algorithms", MIT Artificial Intelligence Laboratory memo 561, Cambridge, Massachussets.

[Marr] Marr, D., Palm, G., Poggio, T., (1978),"Analysis of a Cooperative Stereo Algorithm", Biol. Cybernetics 28,223-239, Springer Verlag, 1978

[Rosen 1] Rosenfeld, A.,Hummel, R., Zucker, S.W., (1976),"Scene Labelling by Relaxation Operations", IEEE Trans. Systems.Man, Cybernetics 6, 1976, 420-433.

[Sabbah] Sabbah, D., (1981) , Ph.D. dissertation, in preperation, University of Rochester, Rochester, N.Y.

[Zucker] Zucker, S.W., (1980),"Labelling Lines and Links: An experiment in Cooperative Computation", TR 80-5, McGill University, Montreal, Canada.