Introduction to Cognitive Science: Notes

II: Representing the World Symbolically

- Readings for this section: Huffman 1971

II: Representing the World Symbolically

- The Marr and Poggio network can be seen as a (partial) model of space itself.
- It could be used as input to guide reaching and grasping, acting as an input to the cerebellum, which represents the learned associations linking visual and motor cortex in actions like grasping a nut or drinking from a cup.
- We will return to this idea in a later section
- But for such actions to be at all useful, the animal has to know that the surfaces in view belong to a nut or a cup, rather than, say, a snake or a tiger.

Object Recognition

- This is a quite different kind of problem from representing the position of a surface in three-space. We need to *recognise* objects as cups, snakes, etc. *anywhere* in space, in order to act *appropriately* on knowledge of where they are.
- This is hard, because:
  - Not all surfaces of the object are visible
  - Even visible surfaces may be obscured by nearer objects
- Marr proposed that object recognition and scene analysis required a representation called the "Two and a half dimensional sketch".

The $2\frac{1}{2}$D Sketch

- The $2\frac{1}{2}$D Sketch is a two dimensional array in which information from many sources, including stereopsis, concerning the orientation and relative depth from the observer of every region in the image is represented, together with information about edges, or abrupt changes in those values.

- We'll assume that this information is *incomplete*. That is, some regions may be too distant for binocular disparity to yield depth, and some edges may be missing or incomplete.

- In particular, finding edges from grey-scale data alone is a very hard problem.

- Nevertheless, as an example of scene analysis using the $2\frac{1}{2}$D sketch, we will consider the information that is available from edges alone, then think about integrating depth and orientation information.

Line Drawings

- We can think of this as a *line drawing*, though in fact it is a neurally represented data structure called a *picture graph*, representing the connectivity between lines representing edges. For example:

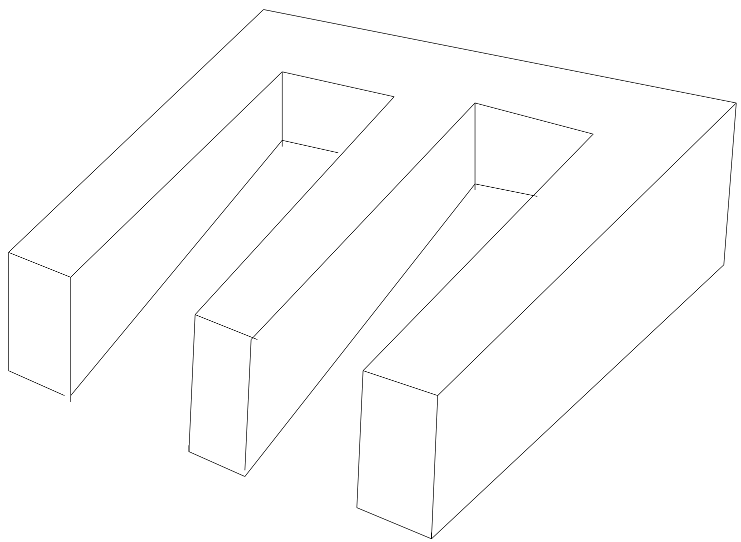

Figure 1: A plane-faced object with trihedral vertices

What's Wrong with This Picture?

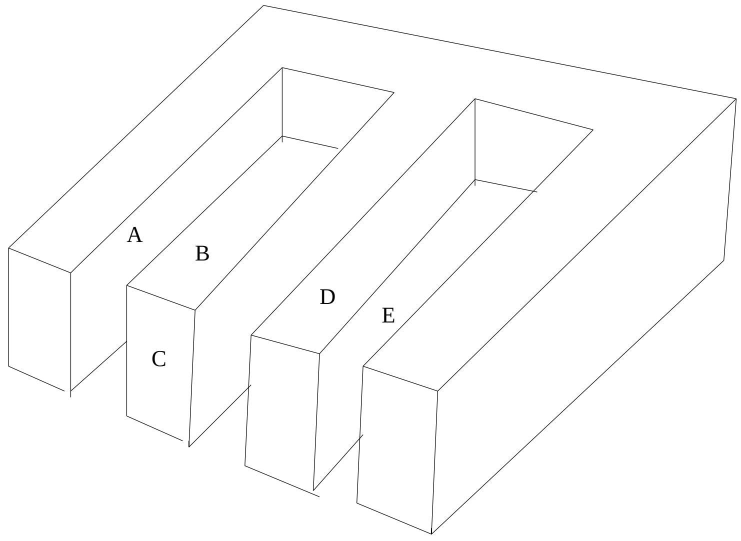

Figure 2: An anomalous object

Guzman 1968

- Local information about connectivity from junction types:

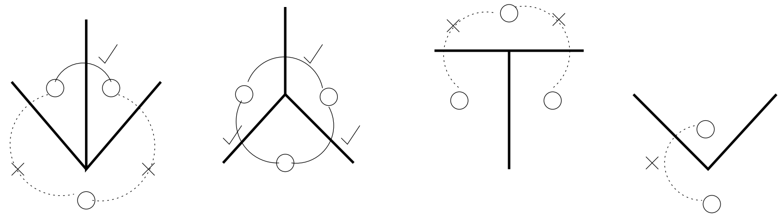

Figure 3: Positive ($\surd$) and negative ($\times$) links (after Guzman).

- *Link Inhibition Heuristic*: a positive link between two regions across a line can only be accepted in the absence of any negative evidence from the junction at the other end of the line.

The General Position Assumption

- There is more information in junctions, provided we assume that the viewpoint is in "General Position", such that a slight change does not change picture topology:

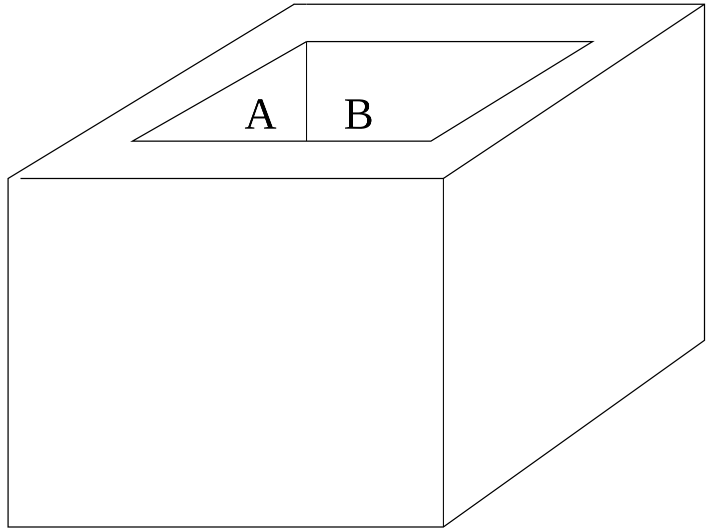

Figure 4: A block with a hole

The General Position Assumption

- An interpretation that violates General Position

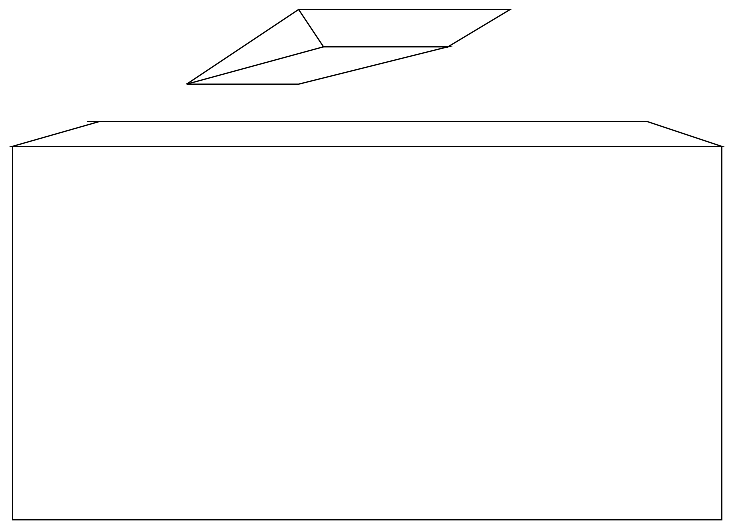

Figure 5: A scene corresponding to a wrong analysis of the previous picture.

Huffman-Clowes Line Labeling

- All possible interpretations of trihedral junctions under the assumption of General Position.

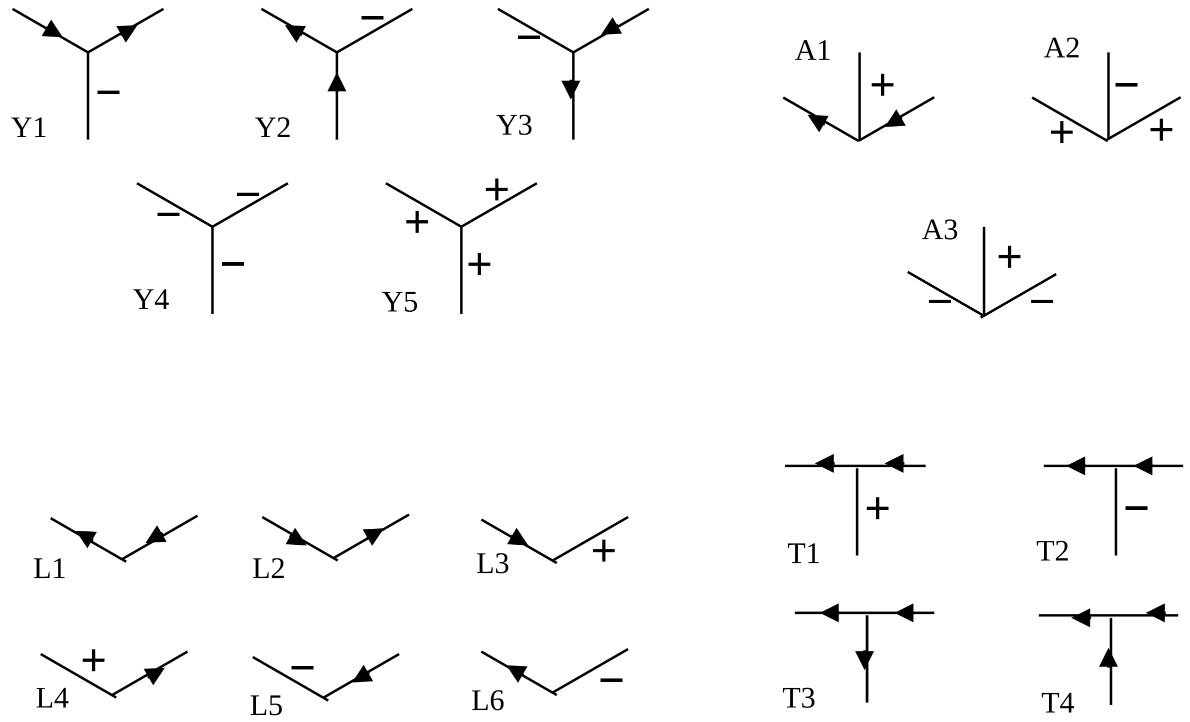

Figure 6: Huffman/Clowes Junction Labels

Holes Correctly Analyzed

- Using this richer representation of junction interpretations, we can recast the Link inhibition Heuristic as a principle of Edge Consistency:

  (1) EDGE CONSISTENCY: Any consistent assignment of labels to the junctions in the picture must assign the *same* line-label ($+$, $-$, $\leftarrow$, $\rightarrow$) to any given line.

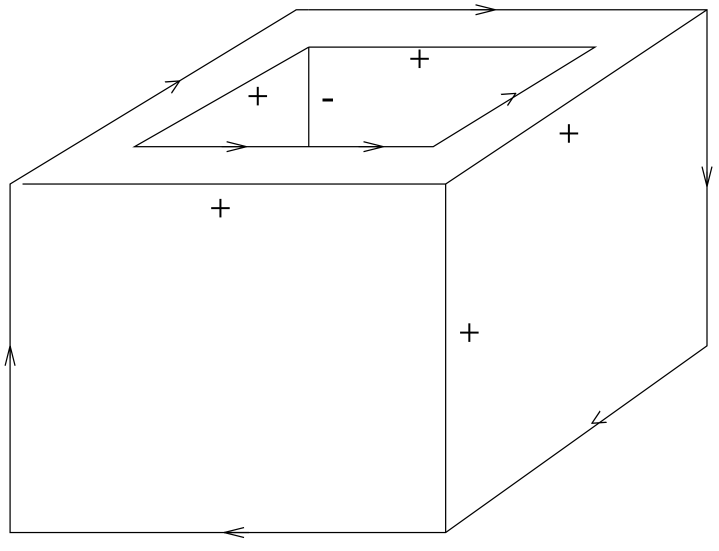

Figure 7: A block with a hole, labeled

Waltz-Mackworth Algorithm

- There are at least three other labelings of 7—what are they, and are they possible interpretations? Are there any *other* labelings?
- To find all and only the possible labelings of a figure we need an algorithm.
- Brute force search is exponential in the average ambiguity of the labels $O(4.5^n)$.
- However, Guzman's observation that most though not all of the information is in *adjacent* pairs of junctions suggest that the branching factor can be cut down by first eliminating all junction labels that are inconsistent with all labels on any neighbor junction.
- This is the idea behind Waltz filtering Waltz 1975.

Waltz-Mackworth Algorithm

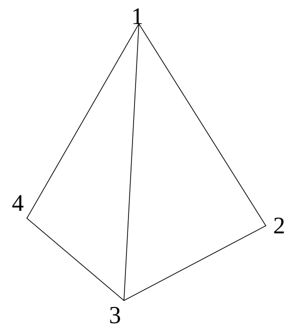

Figure 8: A Minimal Scene

- The label A2 for junction 1 is initially edge consistent with the label L3 on 2, the A2 label on 3, and the L4 label on 4 (although these labels will turn out not to be consistent with each other.)
- However, the label L4 for 2 is not edge-consistent with any label on 1, so it can be removed from the 2's list
- This means that label A2 on junction 3 is not edge-consistent with any label on junction 2, so it can be removed from that junction's list.

Waltz-Mackworth Algorithm

- However, this means that the arrow label of type A2 on junction 1 is no longer edge-consistent with any label on junction 3. *We must therefore make sure that at some point we revisit junction 1, so that A2 will be removed from its list of edge-consistent labels.*

- This example reveals the following important principle:

- *Any algorithm must ensure that, if a label is removed from a junction i, then the labels on all of its neighbours will be reexamined, in case some label has now ceased to be edge-consistent with i.*

- This principle will need one further refinement, but in essence it is the principle behind the algorithms developed by Waltz in his 1972 thesis (cf. Waltz 1975 and Mackworth 1977).

Waltz-Mackworth Algorithm

- The simplest such algorithm (called AC1 for reasons we'll get to) can be stated informally as follows:

  (2) Associate with each junction in the picture a set of possible junction labels for that junction type, as given in figure 6.
  Repeat the following procedure until there is no change to the set of labels associated with any junction:
  For each junction *i* in the picture, for each neighbouring junction *j* in the picture, remove any junction label from *i* for which there is no edge-consistent junction label on *j*

- This simple algorithm is unecessarily costly in that, towards the end of the iteration, most of the junctions will be being repeatedly examined without any change to their label-lists.

- The sources of this inefficiency can be thought of in the following terms.

Waltz-Mackworth Algorithm

- First, at each iteration, only junctions i at least one of whose neighbours j lost a label at the previous iteration need be examined. Such junctions must be examined because the lost label on j may have been the only label on j consistent with some label on i.

- Second, at each iteration, any junction i with a neighbour j that *did* lose a label *still* need not be examined if the reason the neighbour j lost that label was because of edge-inconsistency with i itself. In that case the labels on i must be consistent with the remaining labels on j.

- Third, the only respect in which some label on such a junction i may have been made inconsistent by the removal of a label on a neighbouring junction j is with respect to j itself, since the line-labels imputed to the line (i, j) by the labels on i are the only thing that may have been made inconsistent by removing this particular label. consistent.

Waltz-Mackworth Algorithm

- These observations mean that we should define our algorithm with respect to an independent list of "arcs", where an arc is a *directed* pair of neighbouring junctions, so that there are two arcs (i, j) and (j, i) for each line joining junctions i and j in the picture.
- This implies defining the labels at a junction i in terms of lists of directed arcs represented by ordered pairs of junctions numbers i, j.
- We then define the algorithm in terms of *arc-consistency* rather than edge-consistency, defined as follows for the line-labeling task:

(3) ARC CONSISTENCY: Any consistent assignment of labels to the junctions in the picture must assign the *same* line-label ($+$, $-$, $\rightarrow$, $\leftarrow$) to an arc (i, j) as to the inverse arc (j, i),

Such algorithms are known as "arc-consistency," or AC, algorithms.

Waltz-Mackworth Algorithm

- A version of a more complex but less costly algorithm than the previous one, known as *AC-3*, can then be stated as follows. (A "queue" is a list to which things can only be added at the end).

  (4) *AC-3*: Associate with each junction in the picture a set of possible junction labels for that junction type, as given in figure 6.
  Make a queue of all the arcs (i, j) in the picture and repeat the following procedure until the queue is empty:
  Remove an arc (i, j) from the queue. For each label l on junction i, if there is no label on j which is arc-consistent with l, remove l from i. If any such label is removed from junction i, then for all arcs (k, i) in the picture except (j, i), put (k, i) on the queue if it isn't there already.

Waltz-Mackworth Algorithm

- Initial labels:

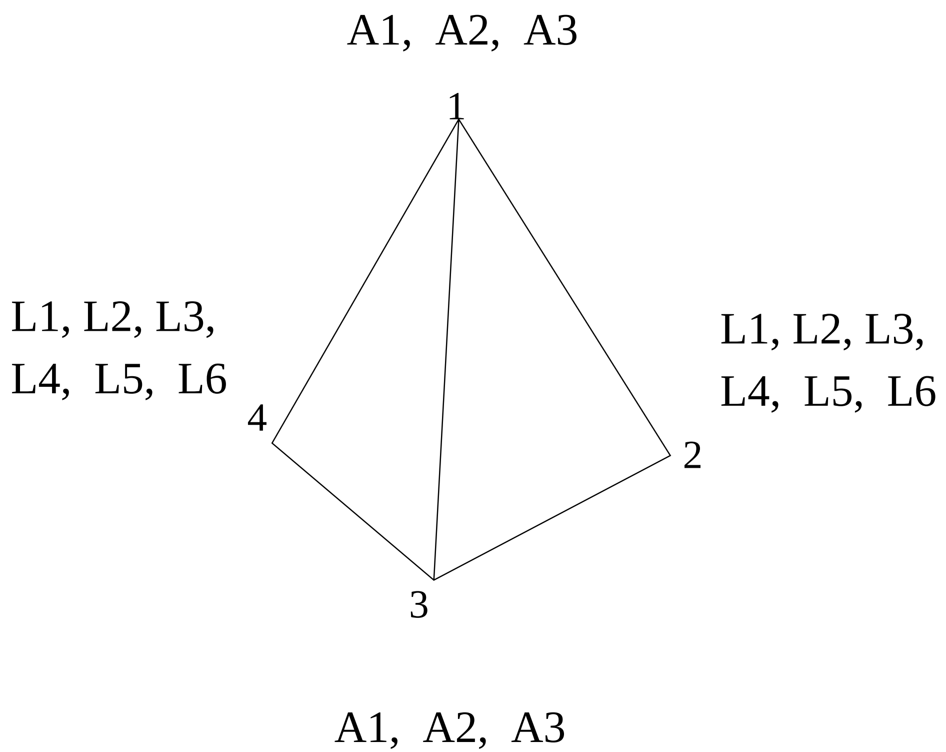

Figure 9: *AC-3* applied to the simple pyramid: State 0.

- We will also assume that the initial queue has the arcs in the following order:

(5) $(1,2)(2,1)(2,3)(3,2)(3,4)(4,3)(4,1)(1,4)(1,3)(3,1)$

- The first arc removed is $(1,2)$. Inspection of the labels in figure 6 corresponding to the indices will show that all labels on 1 are consistent with some label on 2, (although the converse does not apply). No arcs of the form $(k,1)$ are considered for adding to the queue, which is as follows

(6) $(2,1)(2,3)(3,2)(3,4)(4,3)(4,1)(1,4)(1,3)(3,1)$

- The next arc examined is $(2,1)$. This time, two of the labels on junction 2 are not arc-consistent with any label on 1, namely L2 and L4 (see figure 6). All arcs of the form $(k,2)$ except for $(1,2)$ are therefore considered for addition to the queue. However, they are all already present, so the new queue is simply the following:

(7) $(2,3)(3,2)(3,4)(4,3)(4,1)(1,4)(1,3)(3,1)$

- The set of labels on the picture is reduced as in figure 10

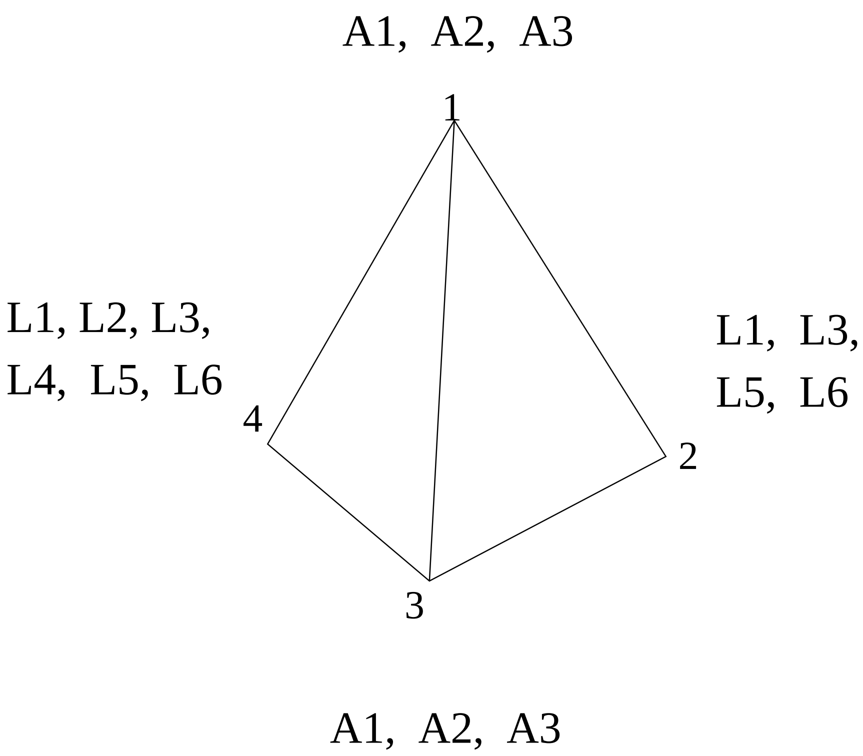

Figure 10: *AC-3* applied to the simple pyramid: State 2.

- The next arc examined is $(2, 3)$. Another label on 2 is not arc-consistent with any label on 3, namely L3, so the labels are as in figure 11.

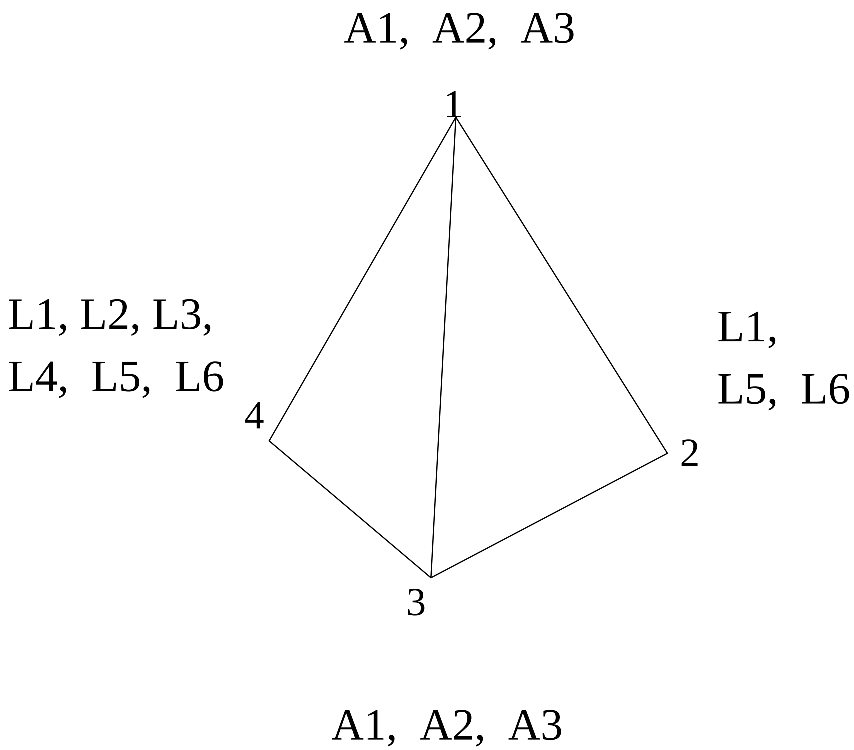

Figure 11: *AC-3* applied to the simple pyramid: State 3.

- Among arcs $(k,2)$ $(1,2)$ is not on the queue, so it goes on at the back. The queue is as follows:

  (8) $(3,2)(3,4)(4,3)(4,1)(1,4)(1,3)(3,1)(1,2)$

- The next arc chosen is $(3,2)$. The label A2 on junction 3 is not arc-consistent with any label remaining on 2, so it is removed. The resulting labeling is as in figure 12.

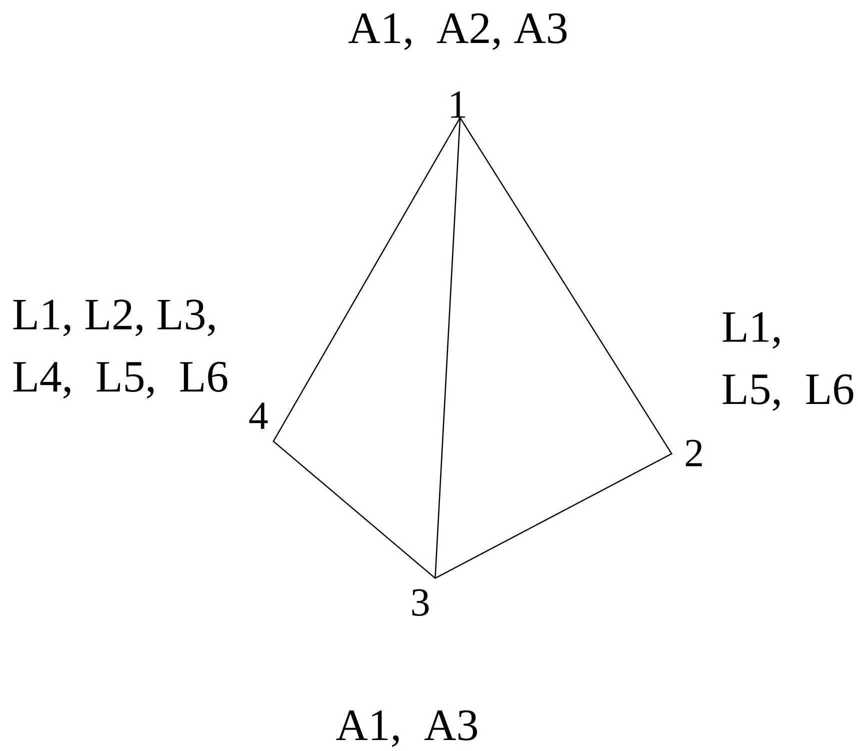

Figure 12: *AC-3* applied to the simple pyramid: State 4.

All the arcs $(k,3)$ other than $(2,3)$ are already on the queue so the following is the next state of the queue:

(9) $(3,4)(4,3)(4,1)(1,4)(1,3)(3,1)(1,2)$

- The remaining steps of the iteration are left as an exercise. They result in the sets of arc-consistent labels for the picture shown in figure 13.

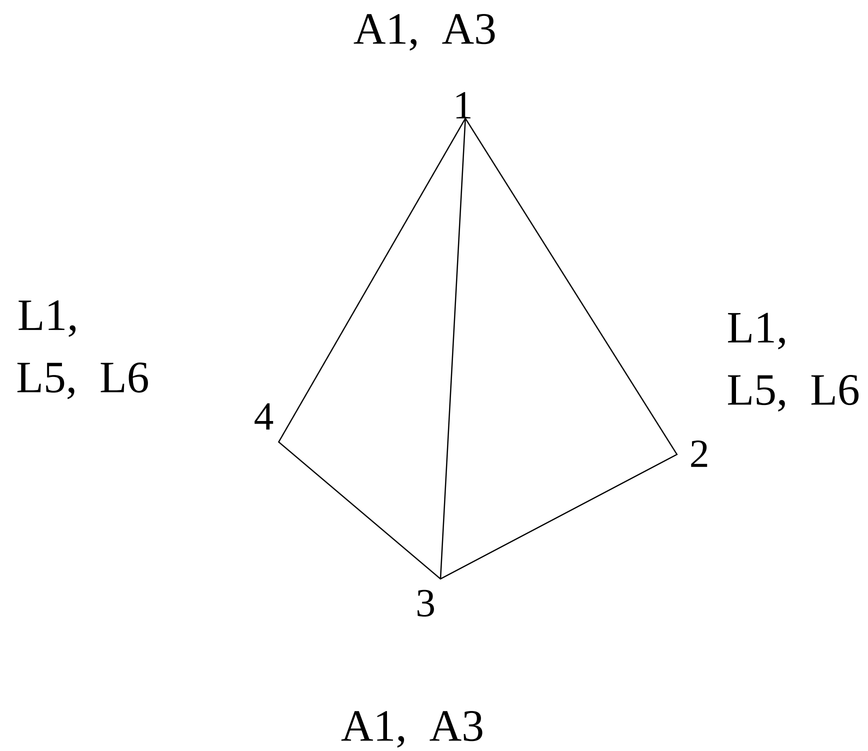

Figure 13: *AC-3* applied to the simple pyramid: Final State.

Waltz-Mackworth Algorithm

- It is important to notice that this does not complete the task of identifying the three interpretations that these labels allow for the figure as a whole. Nevertheless, we have reduced the branching factor for such a search quite considerably.
- Of course, if the arc-consistency algorithm was itself exponential this would not be an interesting result.
- An upper bound on the worst-case complexity of the algorithm is $\mathrm{O}(n^2)$, since if every junction were connected to every other junction in the picture, then there would be $(n * (n-1))$ arcs.
- However, pictures are by definition *planar* – that is, lines cannot cross. The number of lines (and hence arcs) in the picture can only be linear in n.
- All the algorithms considered above (even AC1) are therefore readily parellizable, say as cellular automata of the kind discussed by Marr.

Waltz-Mackworth Algorithm

- This is an important result, because Waltz (1975) discovered that if the set of line and junction labels is generalised to describe more complicated classes of pictures, including polyhedral vertices, and crack and shadow edges, then the benefits of the arc consistency algorithm are even greater in proportion to the complexity of brute-force search than with the simpler set of labels.

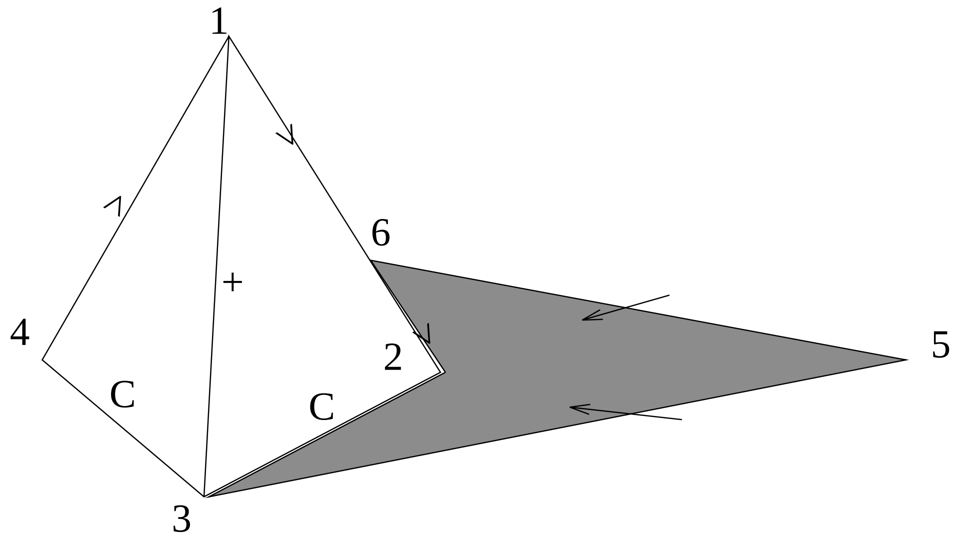

Figure 14: An unambigous picture including shadows and cracks

$2\frac{1}{2}$D Sketch Again

- While Waltz did not consider the use of the kind of orientation information that Marr imputes to the $2\frac{1}{2}$D sketch, This information can be expected to further contribute disambiguation. For example it would completely disambiguate Figures 13 and 7.

- Crucially, Huffman (1971) showed that orientation information *alone* carries the same information as junction connectivity, suggesting that missing lines can be tolerated.

- This theory seems to put us in a position to match the visible aspects of the segmented world to the objects that generate those appearances, say by identifying a structure of *Generalized Cylinders* (Marr and Nishihara 1978; Biederman (1987). This remains an unsolved problem.

- The use of probability to model likely and unlikely interpretations seems not to have been extensively explored in this area (4th Year project).

Transition

- We've seen how the world can be represented by map-like neural structures in the brain, and how symbolic computation can be performed on these structures to segment the world in terms of objects and spatial relations.
- We've also seen a structure that could be used to mediate between such object perception and *actions* like reaching and grasping objects, in the form of the binocular disparity map.
- We've also seen a structure that can be used to interpret visual scenes and support recognition.
- But how do we decide *which* actions to take?

References

Biederman, I, 1987. “Recognition by Components: A Theory of Human Image Understanding.” *Psychological Review* 94:115–147.

Huffman, Donald, 1971. “Impossible Objects as Nonsense Sentences.” In Donald Michie and Bernard Meltzer (eds.), *Machine Intelligence*, Edinburgh: Edinburgh University Press, volume 6. 295–324.

Mackworth, Alan, 1977. “Consistency in Networks of Relations.” *Artificial Intelligence* 8:99–118.

Marr, David and Nishihara, H., 1978. “Representation and Recognition of the Spatial Organization of Three-dimensional Shapes.” *Proceedings of the Royal Society of London* 200:269–294.

Waltz, David, 1975. “Understanding Line Drawings of Scenes with shadows.” In Patrick Winston (ed.), *The Psychology of Computer Vision*, New York: McGraw Hill. 19–91.