INTERPRETIVE VISION AND RESTRICTION GRAPHS

Rodney A. Brooks and Thomas O. Binford

Artificial Intelligence Laboratory, Computer Science Department
Stanford University, Stanford, California 94305

ABSTRACT

We describe an approach to image interpretation which uses a dynamically determined interaction of prediction and observation. We provide a representational mechansim, built on our geometric modeling scheme which facilitates the computational processes necessary for image interpretation. The mechanism implements generic object classes and specializations of models, enables case analysis in reasoning about incompletely specified situations, and manages multiple hypothesized instantiations of modeled objects in a single image. It is based on restriction nodes and quantified variables. A natural partial order on restriction nodes can be defined by comparing the satisifiability of their constraints. Nodes are arranged in an incomplete restriction graph whose arcs represent relations of nodes under the partial order. Predictions are matched to descriptions by finding maximal isomorphic subgraphs of a prediction graph and an observation graph [8] subject to a naturally associated infimum of restriction nodes being satisifiable. In this manner constraints implied by local two dimensional matches of image features to predicted features are propagated back to the three dimensional model enforcing global consistency.

I INTRODUCTION

A. Image Interpretation

A *descriptive* process is one which takes an image and produces a description of image features and their relations found within that image. A *predictive* process is one which uses models of objects expected in an image to predict what features and their relations will be present in the image.

We have constructed a working model-based vision system called ACRONYM [8]. The interaction between prediction and description is stronger than in previous systems. A central aspect of ACRONYM is that it interprets images at the level of three dimensional models.

Here we describe a new layer of representation built on the ACRONYM system. We briefly describe a mechanism for reasoning about incompletely specified geometric situations. All this has been implemented. We describe how matching of two dimensional features can be mapped back to constrain geometric uncertainties in three dimensions in order to obtain a three dimensional understanding of an image. This aspect of the new system is still being implemented (June 1980).

B. The ACRONYM System

ACRONYM itself covers a wider range of tasks than vision (see [4], [10]). The reader is referred to [8] for a complete overview of the system and a description of geometric models based on generalized cones as the nodes of a subpart tree. We give here a brief overview of ACRONYM's vision related modules.

A user interacts with the system via a high level *modeling language* and an *interactive editor* to provide three dimensional descriptions of objects and object classes which are viewpoint independent, and to partially model general scenes. The result is the *object graph*. A *library* provides useful prototypes and a *graphics module* provides valuable feedback to the user.

A rule-based module, the *predictor and planner*, takes models of objects and scenes and produces the *prediction graph* which is a prediction of the appearance of objects expected within the scene. It predicts *observables* in the image over the expected range of variations. It provides a plan, or instructions, for lower level descriptive processes and the matcher to find instances of the objects within the image. The process of prediction and planning is repeated as first coarse interpretations are found, more predictions are carried out, and finer, less ambiguous interpretations are produced.

The descriptive aspect of ACRONYM is currently provided by the *edge mapper* [6] which describes monocular pictures as primitive shape elements (*ribbons*) and their spatial relationships. It is goal-directed and is thus programmed by the predictor and planner. The *observation graph* is the result. As with the predictor and planner, the edge mapper may be invoked many times during the course of an interpretation as finer levels of description become desirable. ACRONYM will incorporate stereo and advanced edge-based description modules (Baker [2] and Arnold and Binford [1]). This will provide three dimensional cues directly.

The *matcher* interfaces description and prediction. It finds maximal subgraphs of the observation graph isomorphic with subgraphs of the prediction graph, which also meet global consistency requirements. In the new implementation the matching process is mapped back to three dimensional models. Such higher level understanding ensures global consistency and enables deductions about three dimensional structures from a single monocular image. The matcher re-invokes the predictor and planner and the edge mapper to extend the two graphs which it is matching in the context of successfully completed submatches. This provides direction to both prediction and description and reduces dramatically the number of possibilities each must consider.

Lowe [9] has implemented a system which determines parameters of models including articulations from correspondences of a match. This module provides predictions from a tentative interpretation to guide detailed verification.

We have concentrated on two classes of images in our development work; aerial images of airport scenes, and scenes of industrial workstations. Together they provide a wide range of challenges and force us to look for general methods and solutions to problems, since they are sufficiently dis-similar that special purpose solutions will fail on one of the two.

II REPRESENTATION

A. Requirements

We have chosen to describe the world to ACRONYM in terms of three dimensional models of objects, and their expected three dimensional spatial relationships ([8] contains details). The representation is a coarse to fine description of objects as subpart trees of generalized cones [3]. In this paper we are concerned with ways to represent variations within models, and how to keep track of multiple inconsistent instances which may arise during image interpretation. Thus what follows does not rely on generalized cones as the representational primitive.

In structured situations the exact dimensions of objects may be known in advance, but their orientation may be uncertain. For instance it may be known that a bin is full of a particular type of part, but the parts may have been dropped in with arbitrary orientations. In an industrial automation domain such descriptions may already be available from a Computer Aided Design data-base. In less structured environments, not even the dimensions of objects will be known exactly. For instance for some aerial image interpretation tasks it is desirable to represent both the class of wide bodied passenger jet aircraft, and particular types of aircraft such as a Boeing 747, and even more particular models such as a 747-B. Thus it is necessary to represent constrained variations in shape, size and even structure (e.g. different aircraft have different engine configurations), and constrained variations in spatial relations between objects. Consider also that an F-111 aircraft can have variable wing geometry. A manipulator arm has even more complex variations in spatial relations between its subparts.

The appearance of an object may change qualitatively, rather than merely quantitatively, over the allowed variations in its size, shape, structure or orientation relative to the camera. Thus it will often be necessary to carry out some case analysis in prediction of appearance, and put further constraints on models, and to keep such mutually confliciting hypotheses in the prediction graph, until such time as they can be confirmed or denied by descriptive processes. The prediction graph represents case as combinations of components instead of explicit enmueration of all cases.

As interpretation of an image proceeds, constraints on the exact values of variations within a model will be derived from the matches made in the image. However there may be multiple instances of a modeled object within the image. Parts on a conveyor belt will have different orientations. Aircraft at a passenger terminal will have different lengths and wing spans. Thus multiple instances of objects must be representable in the interpretation graph.

B. Representing Variations

In the following discussion we will consider the problem of modeling both the generic class of wide-bodied passenger jet aircraft, and specific wide-bodied passenger jet aircraft, such as the Boeing 747, Lockheed L-1011, McDonnell-Douglas DC-10 and the Airbus Consortium A-300. We will then discuss a wider situation where such aircraft are on runways and taxiways, and there are undetermined variables in the camera model.

We need to represent both variations in size (e.g. different aircraft subclasses will have different fuselage lengths), and variations in structure (e.g. different aircraft subclasses will have different engine configurations). In both cases we want to represent the range of allowable variations. We consider the broader problem of quantification of sets. Furthermore, there will sometimes be interdependencies between these variations (e.g. a scaling between fuselage length and wing span).

```
Node: FUSELAGE-CONE
  NAME:               SIMPLE-CONE
  SPINE:              Z0005
  SWEEPING-RULE:      CONSTANT-SWEEPING-RULE
  CROSS-SECTION:      Z0004

        Node: Z0005
          NAME:             SPINE
          TYPE:             STRAIGHT
          LENGTH:           FUSELAGE-LENGTH

        Node: CONSTANT-SWEEPING-RULE
          NAME:             SWEEPING-RULE
          TYPE:             CONSTANT

        Node: Z0004
          NAME:             CROSS-SECTION
          TYPE:             CIRCLE
          RADIUS:           FUSELAGE-RADIUS
```

Generalized cone representation of fuselage.
Figure 1.

The primitive representational mechanism used in ACRONYM is that of units and slots. Objects are represented by units, as are generalized cones, cross-sections, sweeping-rules, spines, rotations and translations to name the more important ones. Figure 1 shows four units with their slots and fillers from a particular ACRONYM model. They describe the generalized cone representing the fuselage of the generic wide-bodied passenger jet aircraft. Note that units are referred to as "Nodes" because they are nodes of the Object graph of figure 1. The NAME slot is a distinguished slot which all units possess. It describes the entity represented by the unit and corresponds to the SELF slot of KRL units [5]. Units identified by "Z" followed by a four digit number are those which were given no explicit identifier by the user who modeled the object. The modeling language parser has generated unique identifiers for them.

The value of a slot is given by its filler. Slot fillers may be explicit, such as "2" or "STRAIGHT". They can also be symbolic constants in the same sense as constants are used in programming languages such as PASCAL. Such fillers are fine for representing specific completely determined objects and situations. Slots may be filled by a *quantifier*, or any evaluable expression involving quantifiers. A quantifier is an identifier with a constraint system (quantification). "FUSELAGE-LENGTH" and "FUSELAGE-RADIUS" are examples of such quantifiers in figure 2.

The following constraints might be imposed upon FUSELAGE-LENGTH and FUSELAGE-RADIUS when modeling the class of wide-bodied passenger jet aircraft:

```
(≤ 40.0 FUSELAGE-LENGTH)  (≤ FUSELAGE-LENGTH 70.0)
(≤ 2.5 FUSELAGE-RADIUS)   (≤ FUSELAGE-RADIUS 3.5)
(≤ 15.0 (QUOTIENT FUSELAGE-LENGTH FUSELAGE-RADIUS))
```

These constrain the range of allowable length and radius, and express a lower bound on the ratio of length to radius.

Quantifiers express allowable variations in dimensions of objects and in the structure of objects. Figure 2 gives the complete subpart tree for a model of generic wide-bodied passenger jet aircraft. For brevity, not all the slots of the OBJECT units are shown here. The QUANTIFIERS slot is explained later. The SUBPARTS slot of an OBJECT unit is filled with a list of subparts giving the next level of description of the object. Entries in the list can be simple pointers to other OBJECT units (e.g. JET-AIRCRAFT has three substructures: STARBOARD-WING, PORT-WING and FUSELAGE). They can also be more complex such as the single entry for the subparts of STARBOARD-WING, which specifies a quantification of subparts called STARBOARD-ENGINE. In this case the quantification is the quantifier F-ENG-QUANT. Note that PORT-WING has a quantification of PORT-ENGINEs as subparts, which is represented by the same quantifier F-ENG-QUANT. This explicitly represents one aspect of the symmetry of the aircraft: it has the same number of engines attached to each wing. Constraints on this quantifier and on R-ENG-QUANT, the number of rear engines might be:

```
(≤ 1 F-ENG-QUANT)  (≤ 2 F-ENG-QUANT)
(≤ 0 R-ENG-QUANT)  (≤ 1 R-ENG-QUANT)
(> 3 (PLUS F-ENG-QUANT R-ENG-QUANT))
```

These say that there must be either one or two engines on each wing, zero or one at the rear of the aircraft, and if there are two on each wing then there are zero at the rear.

Symmetry of size (such as length of the wings) can likewise be represented by using the same quantifier as a place holder in the appropriate pair of slots.

Our complete model for a generic wide-bodied passenger jet aircraft has 28 quantifiers describing allowable variations in size and structure.

```
Node: JET-AIRCRAFT
  NAME:              OBJECT
  SUBPARTS:          (STARBOARD-WING PORT-WING
                        FUSELAGE)
  QUANTIFIERS:       (F-ENG-QUANT ENGINE-LENGTH
                        ENGINE-RADIUS
                        WING-ATTACHMENT ENG-OUT
                        ONE-WING-SPAN
                        WING-SWEEP-BACK
                        WING-LENGTH WING-RATIO
                        WING-WIDTH WING-THICK)

Node: STARBOARD-WING
  NAME:              OBJECT
  SUBPARTS:          ((SP-DES F-ENG-QUANT .
                        STARBOARD-ENGINE))
  CONE-DESCRIPTOR:   STARBOARD-WING-CONE

Node: STARBOARD-ENGINE
  NAME:              OBJECT
  CONE-DESCRIPTOR:   PORT-ENGINE-CONE

Node: PORT-WING
  NAME:              OBJECT
  SUBPARTS:          ((SP-DES F-ENG-QUANT .
                        PORT-ENGINE))
  CONE-DESCRIPTOR:   PORT-WING-CONE

Node: PORT-ENGINE
  NAME:              OBJECT
  CONE-DESCRIPTOR:   PORT-ENGINE-CONE

Node: FUSELAGE
  NAME:              OBJECT
  SUBPARTS:          (RUDDER STARBOARD-STABILIZER
                        PORT-STABILIZER)
  QUANTIFIERS:       (STAB-ATTACH STAB-WIDTH
                        STAB-THICK STAB-SPAN
                        STAB-SWEEP-BACK
                        STAB-RATIO)
  CONE-DESCRIPTOR:   FUSELAGE-CONE

Node: RUDDER
  NAME:              OBJECT
  SUBPARTS:          ((SP-DES R-ENG-QUANT .
                        REAR-ENGINE))
  CONE-DESCRIPTOR:   RUDDER-CONE

Node: REAR-ENGINE
  NAME:              OBJECT
  CONE-DESCRIPTOR:   REAR-ENGINE-CONE

Node: STARBOARD-STABILIZER
  NAME:              OBJECT
  CONE-DESCRIPTOR:   STARBOARD-STABILIZER-CONE

Node: PORT-STABILIZER
  NAME:              OBJECT
  CONE-DESCRIPTOR:   PORT-STABILIZER-CONE
```

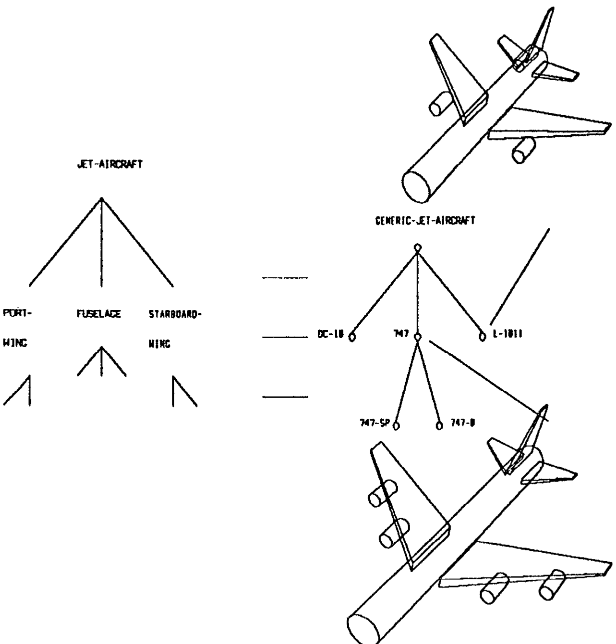
Subpart tree of generic passenger jet.
Figure 3.

C. Representing Classes

It should be clear that to model a subclass of wide-bodied passenger jet aircraft we need only provide a different (more restrictive) set of constraints for the quantifiers used in the general model. To model a specific type of aircraft we could force the constraints to be completely specific (e.g. (= FUSELAGE-LENGTH 52.8))). Thus we will not need to distinguish between specialization of the general model to a subclass, or an individual.

Given that subclasses use different sets of constraints, the problem arises of how to represent multiple subclasses simultaneously. We introduce a new type of node to the representation: a restriction node. These are the embodiment of specialization.

A restriction node has a set of constraints associated with it. If a set of values can be found for all the quantifiers mentioned in the constraints such that all the constraints are simultaneously satisifed, then we say the restriction node is satisifiable. A partial order can be defined on restriction nodes by saying that one restriction node is more restrictive than another if its set of sets of satisfying values is a subset of that of the second node.



Different views of the generic model.
Figure 4.

For the example of the generic wide-bodied passenger jet aircraft the constraints are associated with some restriction node, GENERIC-JET-AIRCRAFT say. To represent the class of 747s a more restrictive node can be included, e.g.:

```
Node: BOEING-747
  NAME:          RESTRICTION
  SUPREMA:       (GENERIC-JET-AIRCRAFT)
  TYPE:          MODEL-SPECIALIZATION
  CONSTRAINTS:   <list of constraints>
```

It is constructed by taking the constraints associated with the GENERIC-JET-AIRCRAFT restriction node, and merging in additional constraints to specialize to a BOEING-747.

A model is always accessed in the context of a restriction node. Thus when reasoning about the generic class of wide-bodied aircraft, the predictor and planner will access the JET-AIRCRAFT model and base its reasoning on the constraints given by the GENERIC-JET-AIRCRAFT restriction node. When reasoning about Boeing 747s it will base its reasoning about the JET-AIRCRAFT model on the constraints given by the BOEING-747 restriction node. Figure 4 conveys the flavor of viewing the JET-AIRCRAFT through different restriction nodes to see different models. (And in fact, the drawings of the two types of aircraft were produced by ACRONYM from the indicated restriction nodes.) In modeling subclasses, restriction nodes typically form a tree rather than a graph.

D. Representing Spatial Relations

Affixments are coordinate transforms between local coordinate systems of objects. They are comprised of a rotation and a translation.

Sometimes affixments vary over an object class. For instance the in generic wide-bodied passenger jet aircraft model the position along the fuselage at which the wings will be attached will vary with particular types of aircraft. Articulated objects are modeled by variable affixments. Variable affixments can also be useful for modeling spatial relationships between two objects - for instance an aircraft is on a runway.

We represent a vector as a triple (a,b,c) where a, b and c are scalars. We represent a rotation as a pair <v,m> where v is a unit vector, and m a scalar magnitude. An affixment will be written as a pair (r,t) where r is a rotation and t a translation vector. We will use some special vectors also: $\hat{x}$, $\hat{y}$ and $\hat{z}$. We use $*$ for the composition of rotations, and $\circledast$ for the application of a rotation to a vector.

In ACRONYM we use the quantifier mechanism to represent affixments which describe a class of coordinate transforms. This gives symbolic representations for rotations and translations.

Consider the problem of representing the fact that an aircraft is somewhere on a runway. Suppose the runway has its x axis along its length, the y axis perpendicular at one end, and the positive z direction vertically upward. Suppose that the coordinate system for the aircraft has its x axis running along the spine of the fuselage and has its z axis skyward for the standard orientation of an aircraft. Then to represent the aircraft on the runway we could affix it with the affixment:

```
(<ẑ, ORI>, (JET-RUNWAY-X, JET-RUNWAY-Y, 0))
```

where ORI, JET-RUNWAY-X and JET-RUNWAY-Y are quantifiers with the following constraints:

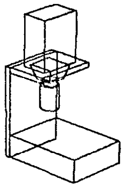
```
(≤ 0 JET-RUNWAY-X)   (≤ JET-RUNWAY-X RUNWAY-LENGTH)
(≤ 0 JET-RUNWAY-Y)   (≤ JET-RUNWAY-Y RUNWAY-WIDTH)
```

Notice that ORI is unconstrained. The aircraft is constrained to be on the runway, in the normal orientation for an aircraft (e.g. not upside down), but it does not constrain the direction in which the aircraft is pointed. If we wished to constrain the aircraft to approximately line up in the direction of the runway we could include a constraint on the quantifier ORI, allowing for some small uncertainty. In general, constraints on a single quantifier may derive from different coordinate systems.

III PREDICTION AND MATCHING

A. Using Constraints

Prediction consists of examining the model in order to find features which are invariant over the range of variations in the model [8], or over a sufficient range to allow a small number of cases. A mixture of reasoning directly about the model, and reasoning about the constraints is needed to find invariants.



Electric Screwdriver and Holder
Figure 5.

Consider the electric screwdriver holder and electric screwdriver in figure 5. This is a display of an ACRONYM model of a tool for the Stanford hand-eye table. The position and orientation (about the vertical axis) are not known. Neither are the exact camera pan and tilt known.

Under these conditions the expression for the orientation of the screwdriver tool relative to the camera, as obtained directly from the model, is:

$$\langle\hat{x},TILT\rangle * \langle\hat{y},(-\ PAN)\rangle * \langle\hat{z},3\pi/2\rangle * \langle\hat{y},3\pi/2\rangle * I * \langle\hat{z},ORI\rangle * I * \langle\hat{y},3\pi/2\rangle * \langle\hat{y},\pi/2\rangle * \langle\hat{y},\pi/2\rangle * I * I * I * I$$

where I is the identity rotation, PAN and TILT are quantifiers associated with the camera orientation, and ORI is the unconstrained orientation of the screwdriver holder about the vertical axis.

We have implemented a set of rules in ACRONYM which simplify such expressions to a canonical form, using identities for re-ordering products of rotations. The details can be found in [7]. The canonical form aids in the detection of invariants. E.g. the above expression is transformed to the equivalent expression:

$$\langle\hat{z},3\pi/2\rangle * \langle\hat{y},TILT\rangle * \langle\hat{x},(+\ PAN\ (-\ ORI))\rangle$$

Consider the problem of predicting the appearance of the cylinder in the image. We outline below the chain of reasoning intended for ACRONYM. (The previous predictor and planner rule set carried out a slightly less general but still powerful line of reasoning. For instance from the knowledge that the aircraft is on the runway ACRONYM deduces its image in an aerial photograph is determined up to a rotation about the vertical plus a translation. The rules necessary for a class of computations including the simple example below will be implemented over summer 1980.)

The left most rotation corresponds to a rotation in the image plane and can be ignored when predicting image shape - i.e. shape is invariant with respect to rotation in the image plane. The right most rotation expression is applied directly to the cylindrical tool. But it is a rotation about the x axis, which is the linear axis of a cylinder in our representation [8] and the appearance of a cylinder is invariant with respect to a rotation about its linear axis. Thus for shape prediction we need only consider:

$$\langle\hat{y},TILT\rangle$$

If TILT is sufficiently constrained (as in this example) it may be possible to predict the shape directly. The prediction takes the form of expected image features, their relations, and what constraints local matches to such features produce on the three dimensional model. See section III-C for an example. But note here that the prediction is a conjunction of expected features.

B. Adding Constraints

If TILT in the above example is not sufficently constrained there may be more than one qualitatively different shape possible for the cylinder (e.g. an end view of a cylinder is quite different from a side view). If so it is necessary to make a disjunction of predictions. Note however, that all views need not be explicitly expanded - they can still share much structure.

Each prediction is associated with a new, more restrictive restriction node. It is obtained by adding a new constraint which restricts the model sufficiently to necessitate only a single prediction. Figure 5 gives an indication of the structure of a local prediction, with two different cases considered. Not indicated in that diagram are arcs between the feature predictions which specify relations which should hold between instances of those features in the image.

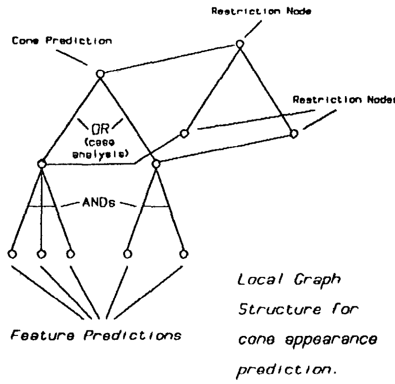


Figure 5

C. Generating Restrictions During Matching

In this section we give an example of an image prediction which generates restriction nodes during matching. We predict the appearance of a ribbon generated by the fuselage of figure 1 in an aerial image. A prediction says what a match of a local feature must imply about both the observed feature, and the aspect of the object whose image generated that feature. Checking this implication (deciding whether the new restriction node is satisfiable) provides an acceptance test for a hypothesized match.

For a projective imaging system the observed distance m between two points distance l apart on the ground is given (approximately) by:

$$m = \frac{cl}{h}$$

where c is a constant dependent on the focal distance of the camera and h is the height of the camera above ground. Thus if the camera is at height HEIGHT (a quantifier) and M1 and M2 are the measurements obtained for length and width from a hypothesized match of a ribbon in the observation graph, then the following constraints must hold if the match is correct:

```
1. (= M1 (QUOTIENT (TIMES CAM-CONST FUSELAGE-LENGTH)
                   HEIGHT))
2. (= M2 (QUOTIENT
          (TIMES 2 CAM-CONST FUSELAGE-RADIUS)
          HEIGHT))
```

In general M1 and M2 will not be given exactly by the observation graph, rather an interval estimate will be supplied. Thus they can be represented by quantifiers with constraints on their upper and lower bounds. If CAM-CONST is known in advance its numeric value can be substituted into the constraints generated by the match. If it too is a quantifier, then it is just one more constrained unknown in the system. At the time of hypothesizing a match, a new restriction node is generated by adding constraints 1 and 2 to the const aints of the restriction node associated with the prediction (see figure 5). If the new node can be shown to be unsatisfiable then the match is rejected.

The following is not meant to indicate a proposed reasoning chain for ACRONYM. Rather it is illustrative of how constraints can imply that a hypothesized match is incorrect. Suppose for some hypothesized match, where CAM-CONST is known to be 100.0, the observed M1 lies between 4.0 and 5.0. Then given the constraints on the fuselage size of section II-B, the height of the camera must be between 800.0 and 1750.0 due to constraint 1 above. If this is inconsistent with a priori constraints on HEIGHT the match can be rejected. In fact a priori constraints on HEIGHT may also put further restrictions on the possible range for FUSELAGE-LEGNTH. Similarly measurement M2 and constraint 2 above will lead to restrictions on HEIGHT. If these restrictions are inconsistent with the 800.0 to 1750.0 bounds already obtained the match should be rejected.

D. From Local to Global

After each phase of local matching, the matcher combines local matches into more global interpretations. This involves finding consistent subgraphs of matches. Previously consistency has only concerned the existence of arcs describing relations between matched ribbons. With the introduction of constraints on quantifiers during the ribbon matching process, these too must be checked for consistency.

Constraints on a quantifier at different HYPOTHESIS-MATCH restriction nodes may actually refer to different quantities in the scene. For instance each potential match for an aircraft may have constraints on FUSELAGE-LENGTH and on HEIGHT. When combining the matches for aircraft to produce an interpretation of the image, there is no reason to require that the constraints on FUSELAGE-LENGTH at these different nodes be mutually consistent. Different instances of wide-bodied passenger jet aircraft will be different lengths. However all the constraints on HEIGHT should be mutually consistent, as there is only one HEIGHT of the camera.

Sometimes when constraints on quantifiers actually correspond to different quantities in the world, it may be that these quantities should have the same value. For instance the ENGINE-LENGTH for the port and starboard engines correspond to physical measurements of different objects in the world. However since aircraft are symmetric the constraints given by the matches on possible values of ENGINE-LENGTH for each engine should be consistent. Thus when clumping the local matches for an aircraft the ENGINE-LENGTH constraints from each submatch should be checked for consistency. If they are not consistent the particular set of local matches should be rejected as inconsistent.

A slot is provided in object units to represent which quantifiers matched at a lower level should be held consistent for interpretation of an object. This is the QUANTIFIERS slot as shown in figure 2. As the matcher is combining local matches it looks up the subpart tree. Any quantifier mentioned in a QUANTIFIERS slot of any ancestor of the object has its constraints copied into the restriction node for the new more global node. As each constraint is introduced it is checked for consistency. This process is not quite straightforward. Sometimes a constraint on a quantifier involves another quantifier which is not being brought into the new match. Such is the case of FUSELAGE-LENGTH and HEIGHT in the example of the previous section when a global interpretation is being made involving many aircraft. Each aircraft provides a constraint on HEIGHT, but each is in terms of the instance of FUSELAGE-LENGTH of the individual aircraft. One solution is to generate a new unique identifier for the quantifier which is not to be constrained by new constraints imposed. Its role is to ensure the continued satisifiability of the local match in light of new global constraints on other quantifiers involved in that match. Other solutions exist, which may result in constraint inconsistencies being missed in return for much simplified constraint analysis.

IV REMARKS

We have described a single part of ACRONYM and have ignored many important issues involved in the construction of a vision system based on the representations given. In particular we have not discussed the analytic power necessary to decide whether constraint sets are satisfiable. We believe that quite weak analytic methods can lead to powerful interpretive capabilities even though they fail to detect large classes of inconsistencies. Nor have we described in detail the methods to carry out the necessary geometric reasoning. These have been discussed in [7] which includes explicit rules for symbolic geometric reasoning in states of uncertain knowledge.

We have provided a representational scheme which facilitates the computational processes necessary for interpretation. The scheme uses restriction graphs to provide specializations of models, to enable case analysis in reasoning about incompletely specified situations, and to manage multiple hypothesized instantiations of modeled objects in a single image.

ACKNOWLEDGEMENTS

This research was sponsored by ARPA contract MDA-903-76-C-0206 and NSF contract DAR-7815914. Support was also provided by the ALCOA foundation.

REFERENCES

[1] Arnold, R. David and Thomas O. Binford, "Geometric Constraints in Stereo Vision," Proc. SPIE Meeting, San Diego, July 1980.

[2] Baker, H. Harlyn, "Edge Based Stereo Correlation," *Proc. ARPA Image Understanding Workshop*, Baltimore, Apr. 1980, 168-175.

[3] Binford, Thomas O., "Visual Perception by Computer," Invited paper at *IEEE Systems Science and Cybernetics Conference*, Miami, Dec. 1971.

[4] Binford, Thomas O., Proc. NSF Grantees Conference, Cornell Univ., Sep. 1979.

[5] Bobrow, Daniel G. and Terry Winograd, "An Overview of KRL, a Knowledge Representation Language," *Cognitive Science* 1, 1977, 3-46.

[6] Brooks, Rodney A., "Goal-Directed Edge Linking and Ribbon Finding," *Proc. ARPA Image Understanding Workshop*, Palo Alto, Apr. 1979, 72-78.

[7] Brooks, Rodney A. and Thomas O. Binford, "Representing and Reasoning About Partially Specified Scenes," *Proc. ARPA Image Understanding Workshop*, Baltimore, Apr. 1980, 95-103.

[8] Brooks, Rodney A., Russell Greiner and Thomas O. Binford, "The ACRONYM Model-Based Vision System," *Proc. of IJCAI-79*, Tokyo, Aug. 1979, 105-113.

[9] Lowe, David, "Solving for the Paramters of Object Models from Image Descriptions," *Proc. ARPA Image Understanding Workshop*, Baltimore, Apr. 1980, 121-127.

[10] Soroka, Barry I., "Debugging Manipulator Programs with a Simulator," to be presented at CAD/CAM-8, Anaheim, Nov. 1980.