Object Recognition – Interpretation Tree

CMPE 264: Image Analysis and Computer Vision
Hai Tao
Object recognition

- Model-based object recognition
  - Given a database of object models, and an input image, model-based object recognition addresses two problems
    - Identification: which models in the database match the data in the image
    - Location: Given that an object in the image matches a given model, what is the location of the object
Four types of problems

Depending on whether there are multiple models and whether there are multiple instances of a particular model appearing in the image, the object recognition problem can be further classified into the following four categories:

- What objects are we looking at (types and number of objects are unknown)?
- What is this part of the image (the type of the object is unknown, one instance in a particular image region)?
- Are there any instances of object X in the image? (object type is known, number and locations of the objects are unknown)
- Is this part of the image an instance of X (object type and location are known)?
Interpretation tree

- Feature based identification method based on early AI algorithms
- Assumptions
  - All image features of a given class have been detected to form a list of symbolic descriptors
  - A list of symbolic descriptors covering all the features of a given object model has been formed
  - The symbolic descriptor of image and model features are expressed in the same format and can be compared directly
  - All descriptors are geometric
- Problem statement
  - Given
    - the list of feature descriptors from a given object model
    - the list of feature descriptors detected in a range image
    - A list of geometric constraints that model features must satisfy
  Find a mapping between model features and image features such that the features match correctly and satisfy the geometric constraints
Interpretation tree

Example

- Each feature is a surface patch described as a rectangle, a square, or a L-shape
- A list of features are formed in the model (right image)
- A list of features are found in the range image (left image)
- The geometric constraint is that features match only if they are the same type

Figure 10.2  (a) Ideal range view of a simple object, showing the labels ($f_i$) of image features (surface patches). $m_1, m_5, m_8$ are rectangular; $m_3, m_6, m_7$ are square; $m_2, m_4$ are L-shaped. (b) Visualization of a surface-based model of the same object, showing the labels ($m_j$) of the object features.
Interpretation tree

- Starting from each feature in the range date, we can search for solution systematically. However, this is a combinatorial problem and sometimes prohibitively expensive to compute.

![Interpretation tree](image)

> **Figure 10.1** An interpretation tree representing the complete search space for a problem involving two image features, \( \{f_1, f_2\} \), and four model features, \( \{m_1, \ldots, m_4\} \). Each level represents all possible matches between an image feature \( f_i \) and a model feature \( m_j \). Therefore, the node \( m_j \) on level \( i \) represents the match \( (f_i, m_j) \).

- Interpretation tree is a depth-first search algorithm for model matching.
Interpretation tree - illustration

- **Depth-search and verification**

  Figure 10.2 (a) Ideal range view of a simple object, showing the labels \( f_i \) of image features (surface patches). \( m_1, m_2, m_8 \) are rectangular; \( m_3, m_6, m_7 \) are square; \( m_2, m_4 \) are L-shaped. (b) Visualization of a surface-based model of the same object, showing the labels \( m_j \) of the object features.

- A solution is verified by back-projecting the model onto the image plane and checking whether model features are close to the image features.

Figure 10.3 Portion of the interpretation tree for the problem of Figure 10.2 expanded when reaching the first interpretation, \( I_I \) (solid path). The figure shows also the path (dashed) corresponding to the correct interpretation, \( I_R \) (but not the nodes generated between \( I_I \) and \( I_R \)).
Interpretation tree algorithm

- Wild cards: fictitious features that matches any image features for which no real matches are found. The purpose of wild card features in the model is to accommodate spurious features in the image.

- The interpretation tree algorithm
  - A depth first search algorithm
  - Verification for dealing with local matches
  - Wild cards for accommodating spurious image features

---

**Algorithm INT_TREE**

Let

- Open the ordered list of nodes to expand, left to right;
- Interp the list of consistent matches forming an interpretation;
- Maxsize the maximum number of nonwildcard matches in any interpretation found;
- W the wild card;
- size(Interp) the number of nonwildcard matches in Interp;
- root a label for the tree root;

consistent(X), X a non-wildcard match, is true if X is consistent with constraints or X=root, and false otherwise. Let [] indicate an empty list, and [a, b, ..., n] a list of elements. Notice that the wildcard is, by definition, consistent with any constraints.

Open = [root], Interp = []; Maxsize = 0;
WHILE (open ≠ [])
BEGIN
  remove leftmost match, X = (fj, mj), from open;
  IF (consistent(X) AND
      max possible size of interpretation on this path ≥ Maxsize)
  BEGIN
    add X to Interp;
    IF (leaf node reached on this path)
    BEGIN
      verify Interp;
      IF (verification succeeds)
      BEGIN
        save Interp;
        Interp = [];
        Maxsize = size(Interp);
      END
      return(failure);
    END
    ELSE (* not leaf, but consistent *)
    let L = (fj+1, mj+1)...
    be the expansion of the current node;
    drop model features already matched from L;
    add L to the left of Open;
    END
    ELSE (* inconsistent match *)
    continue;
  END
END