Aspect Graph Matching

CS510

Lecture #25

April 24, 2002

For Friday, read section 11.2 of Trucco & Verri

Generic Interpretation Tree Algorithm (Part 1)

Note: this algorithm is not an exact match to the one in your book

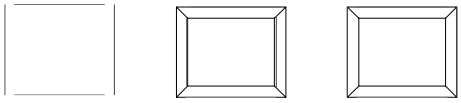
- Let Model be the list of model features $\{m_1,\ldots,m_n\}$
- Let Data be the list of image features $\{d_1,\ldots,d_m\}$
- Let Interp an (initially empty) list of model/data pairs
- Let UnaryP(m_i,d_j) return true iff d_j meets m_i's unary constraints
- Let BinaryP(m_i,d_j,Interp) return true iff the pair (m_i, d_j) is consistent in terms of binary constraints with every pair already in Interp
- List operators (emptyp, destructive pop, non-destructive append)

Generic Interpretation Tree Algorithm (Part 2)

```
InterpTree(Model, Data, Interp) {
  if (emptyp(Model)) return Interp;
  m := pop(Model);
  maxlist = Interp;
  for d in Data do {
    if (UnaryP(m,d) and BinaryP(m,d,Interp)) {
      newlist = InterpTree(Model, Data, append((m,d),Interp));
      if (size(newlist) > size(maxlist)) maxlist := newlist;
    }}
  newlist = InterpTree(Model, Data, Interp);
  if (size(newlist) > size(maxlist)) maxlist := newlist;
  return newlist; }
```

Global Failures

- It is possible to find a locally consistent match that is incompatible with any acceptable geometric transformation

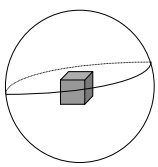


Source Data | Model with Globally Inconsistent Interpretation | Model with Globally consistent Interpretation

Note that inconsistent interpretation satisfies all pairwise parallel and orthogonal relations, but is non-sensical if the object is opaque

The Viewing Sphere

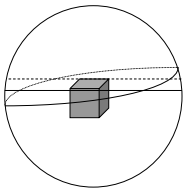
- One way to combat this problem is with *aspect graphs*
- But first, we need to introduce the viewing sphere:



The viewing sphere represents all possible viewpoints looking at an object from a fixed radius

The Viewing Sphere (cont.)

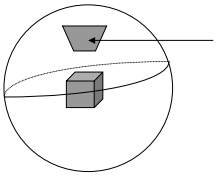
- For any given feature, there is a region on the viewing sphere from which it is visible



The top surface (all four of its lines) can be seen from this region

Aspect Graphs

- An aspect graph segments the viewing sphere into regions
- Inside each region ("aspect"), the same set of features are visible



One aspect. Any feature visible from any point in this region is visible from every point in this region.

Matching with Aspect Graphs

- Aspect graphs can eliminate inconsistent matches:
  - Every model feature is consistent with a set of aspects
    - The regions it can be seen from
  - So pass the set of consistent aspects as an argument to the matching algorithm
  - If the current model line is inconsistent with the currently aspects, skip it
  - Otherwise, for every model/data pair, intersect the aspects with the set of aspects consistent with the model line

Aspect Matching Algorithm

```
InterpTree(Model, Data, Interp, Aspects) {
  if (emptyp(Aspects)) return {};
  if (emptyp(Model)) return Interp;
  m := pop(Model);
  maxlist = Interp;
  if (IntersectP(Aspects, ModelAspects(m))) {
    for d in Data do {
      if (UnaryP(m,d) and BinaryP(m,d,Interp)) {
        newlist = InterpTree(Model, Data, append((m,d),Interp),
                             Intersect(Aspects, ModelAspects(m)));
        if (size(newlist) > size(maxlist)) maxlist := newlist;
  }}}
  newlist = InterpTree(Model, Data, Interp, Aspects);
  if (size(newlist) > size(maxlist)) maxlist := newlist;
  return newlist; }
```

Geometric Matching

- What makes visual interpretation different from other AI-style matching problems?
  - Geometry!
- Every model/data pair restricts the space of possible geometric transformations
- The aspect matching algorithm uses this to prune the search tree

Geometric Matching (cont.)

- Similarity transforms (in-plane rotation, translation, scale) are uniquely determined by three point matches.
- Perspective transformations are uniquely determined by four point matches
  - remember your morphing assignment…
- As soon as the current match fixes the transformation, *you can abandon tree search!*
  - Calculate the transform; it should map model features (almost) directly onto data features

Geometric Matching (redux)

- Perform interpretation tree matching
  - stop whenever the current interpretation defines the geometric transformation
    Note that if model features are matched to NULL, this may occur at uneven depths in the tree
  - Compute the transformation
  - Apply the transformation to unmatched model features
    - Compute their distance (error) to the nearest data feature; include those within a distance threshold in the interpretation
- *What is the complexity of point matching under perspective transformations?*