Data Association, Part II

- Associating single observations at a time often doomed.
  - Eiffel towers are great, but how distinctive is a 90° corner?!

- Approach
  Use multiple observations simultaneously to disambiguate!
  - One corner is weak evidence, but 6 at a time is pretty distinctive!
  - Analogy: recognizing stars vs. constellations.

- Methods (outline)
  - Maximum Likelihood data assoc.
  - TCBB (& IPSC?)
  - SCGP
  - RANSAC
Given \( N \) observations of \( F \) features, which data associations to use?

Idea: if I knew the data assoc, I could solve the \( \text{SLAM} \) problem by computing the posterior distribution. This would be:

\[
p(x | z, a) \quad \text{state} \uparrow \text{assocs.} \quad \text{obs.}
\]

(An aside, we used to assume \( a \) and just write \( p(x | z) \). By Bayes' rule, \( p(x | z) \propto p(z | x) \), which is precisely what our least squares optimization uses as an objective function. It's important that you see this!)

An approach: Just try every value of \( a \), compute \( p(x | z, a) \) and select the \( a \) that maximizes the posterior probability of the maximum likelihood solution.

> how many values of \( a \)? \( F^N \)

> can we search efficiently?

Yes - an "interpretation" tree
The tree terminates at level N with $F^N$ leaves.

- At each leaf, compute posterior soln. (ML).
- Pick the data assoc. that maximizes $p(ML\ sol)$ (equiv., minimizes $\chi^2$ error).

Can we compute $\chi^2$ at intermediate nodes?

Yes! Corresponds to ignoring observations "below".

**Critical** $\chi^2$ cost of a child always $\geq \chi^2$ cost of parent.

This yields pruning heuristic:

If you have a leaf with cost $\chi^2_{best}$, don't expand any nodes with a greater cost.

(equiv. to $A^*$ with a heuristic function of zero.)
How do we handle new features?

→ easy! Consider \((F+1)\) possibilities for each observation.

Uh-oh! What's the ML solution?

→ every observation is a new landmark!

How to fit this?

Attach a cost (or probability) to new landmarks.

\[
p(\tilde z_i \text{ is new}) = K. \quad \text{(for example.)}
\]

→ This penalizes new features, "encouraging" loop closures,

→ Picking \(K\) can be hard.
JCBB addresses the "new landmark" problem by reformulating the problem entirely:

- Instead of ML data assoc, we find the largest set of data assoc such that $\chi^2 < \text{threshold}$.
- Branching factor still F+1, but "null" hypotheses are unfavored.

"Compatibility score" same as ML, except relative to the initial $\chi^2$ at the root of the tree.

```
best = \emptyset \leftarrow \text{our best data assoc},
JCBB(1, graph, obs, \emptyset);
JCBB(depth, graph, obs, h)
  if depth = |obs| (leaf?)
    if |h| > |best|
      best = h
    return
  for i = 1 : F
    Cost = compute cost (graph, obs, h u \{Z \rightarrow f_i\})
    if cost < chi2(\text{DOF}) * k
      JCBB(depth+1, graph, obs, h u \{Z \rightarrow f_i\})
  JCBB(depth+1, graph, obs, h) \text{ (handle F+1 null hypothesis)}.
```

As we add more data assoc, we expect an increase in $\chi^2$ — even if those assoc are correct. The chi2(\text{DOF}) threshold accounts for this natural increase, where \text{DOF} = \# of hyperplanes resulting from associations.
TCBB optimizations

1. Do a cheap compat. test before computing the expensive test.

2. Prune if \( |\text{obs}| - \text{depth} + |h| < |\text{best h}| \)

   \( \text{max # of assocs that we could still make} \)

3. Don’t compute the posterior.

   Instead, \( X^2 = \Delta z^T (HPH^T + GUG^T)^{-1} \Delta z \)

   \( \text{disagreement between predicted value & obs. value} \)

   \( \text{Uncertainty in obs} \)

   \( \text{Uncertainty in prior} \)

   \( \text{Uncertainty in } \Delta z \)

   \( \text{Mahal distance of } \Delta z \text{ WRT. total noise/uncertainty.} \)

   Note: Size of this matrix grows as we descend tree

   \( \Rightarrow O(\text{depth}^3) \) cost to compute compatibility.

4. Use incremental matrix inversion to reduce \( O(\text{depth}^3) \)

   to \( O(\text{depth}^2) \).

Pruning & these tricks can help a lot, but there are still an exponential # of leaves — can be very slow.