Scalable Active Matching

Ankur Handa, Margarita Chli, Hauke Strasdat, Andrew J. Davison

Imperial College London   erc

1. Summary

**How can we achieve truly robust dense, real-time matching?**

The strong priors generally available on image correspondence problems, are often used *partially* and *post-hoc* to resolve matching consensus using algorithms like RANSAC often compromising accuracy for speed. Instead, fully integrating these priors in an 'Active Matching' (AM) approach achieves truly robust performance with rigorous decisions driven by Information Theory. However, the overhead involved in maintaining the joint PDF over feature locations in AM severely limits its scalability. With the aim of bringing the robustness of AM into dense real-time matching, in this paper we present CLAM and SubAM which exhibit dramatically more scalable operation via relaxations to the rigid probabilistic model of AM.

2. Active Matching (AM) [Chli, Davison ECCV 2008]

- Step-by-step search for global consensus in the presence of ambiguity.
- Uses a mixture of Gaussians to describe matching state.
- Search completion defined in terms of probabilistic criteria.

In every matching step:
- **PREDICT** which candidate gives max. MI efficiency
- **MEASURE** this candidate
- **UPDATE** the mixture of Gaussians
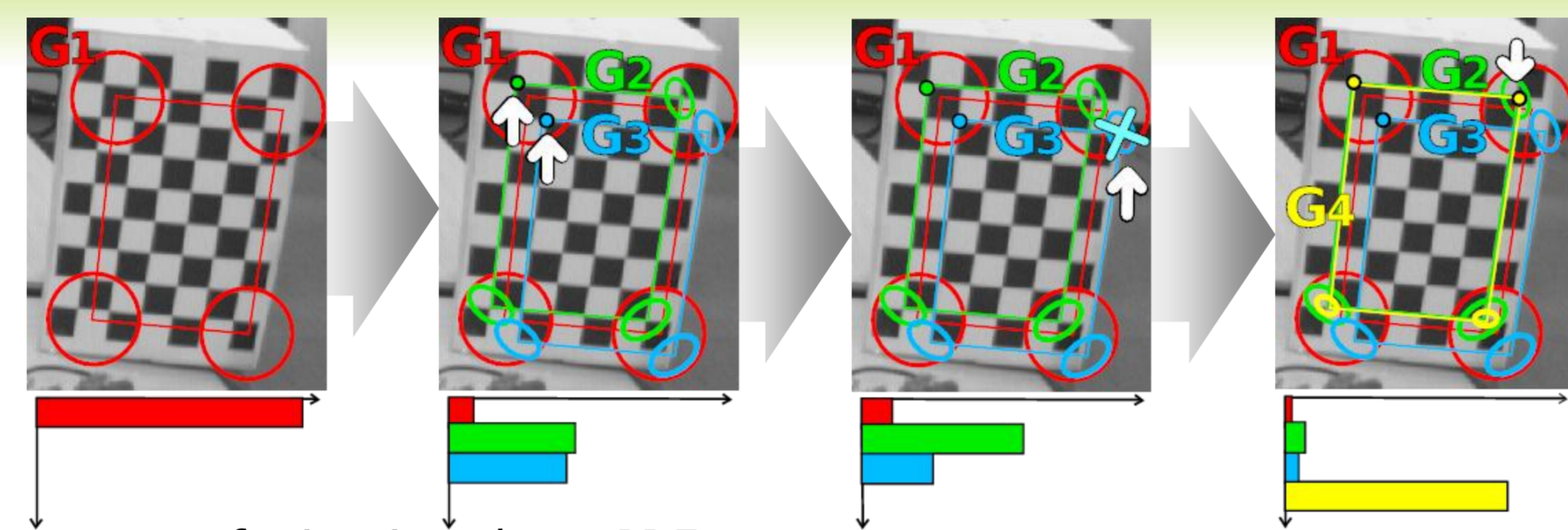
- Highly robust to outliers.
- **Main bottleneck**: Poor scaling to large numbers of features due to the costly maintenance of priors in a dense PDF.
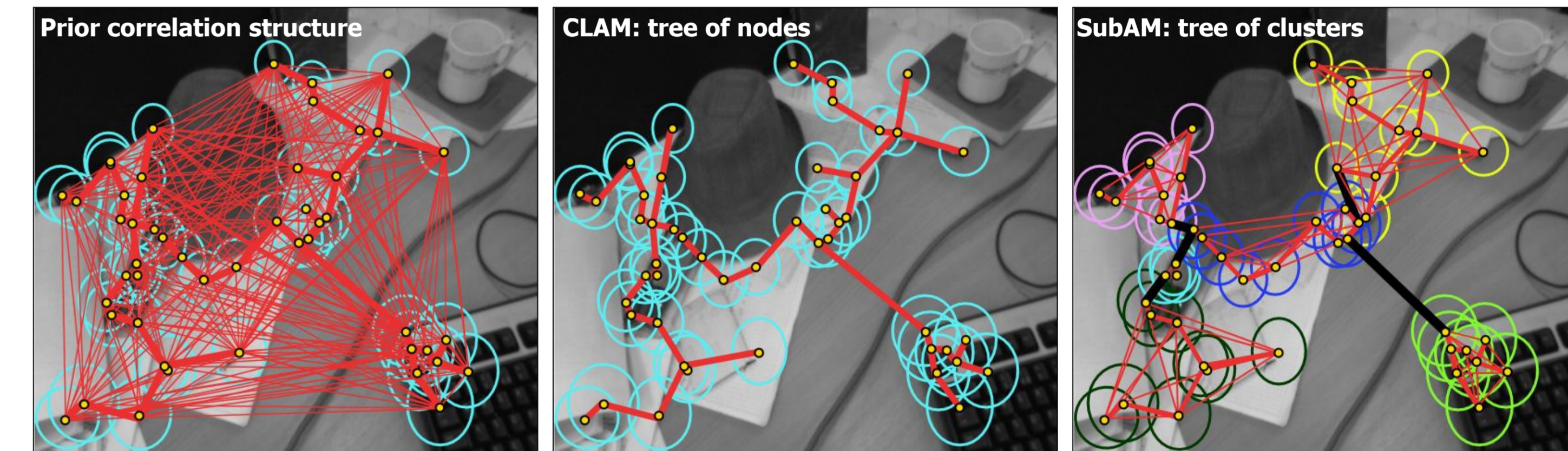

2. Effective Approximations of Matching Priors

- Dense, real-time matching: RANSAC is the dominant, if not imperative choice but sacrifices valued cues for the sake of speed ⇒ **aim:** bring the robustness of AM into dense real-time matching.
- **Pairwise Mutual Information** (MI): an absolute, normalised measure of correlation.
- **[Chow, Liu 1968]**: The optimum approximation of a joint PDF with a 1st order dependency tree is the max. spanning tree in the MI graph.
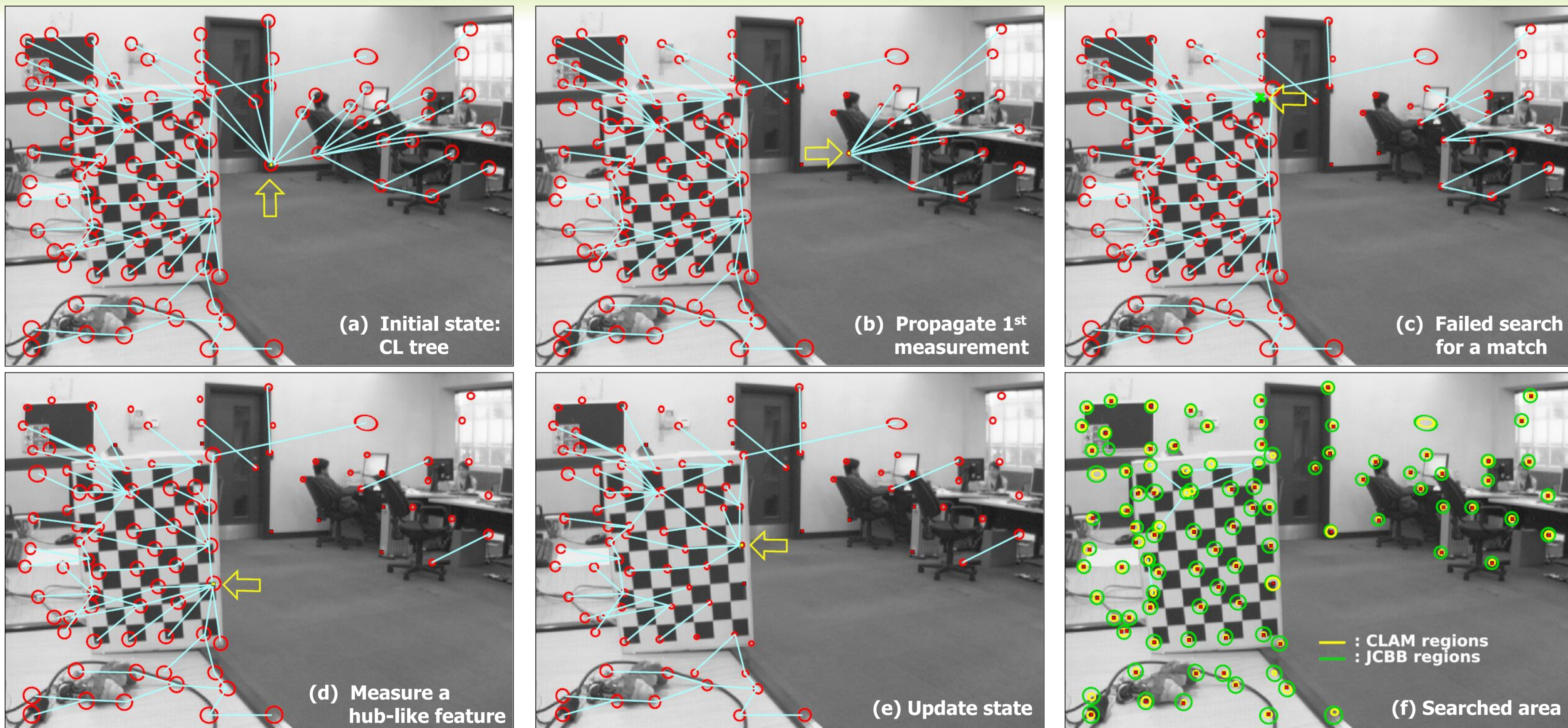

Complete MI graph   Chow-Liu tree
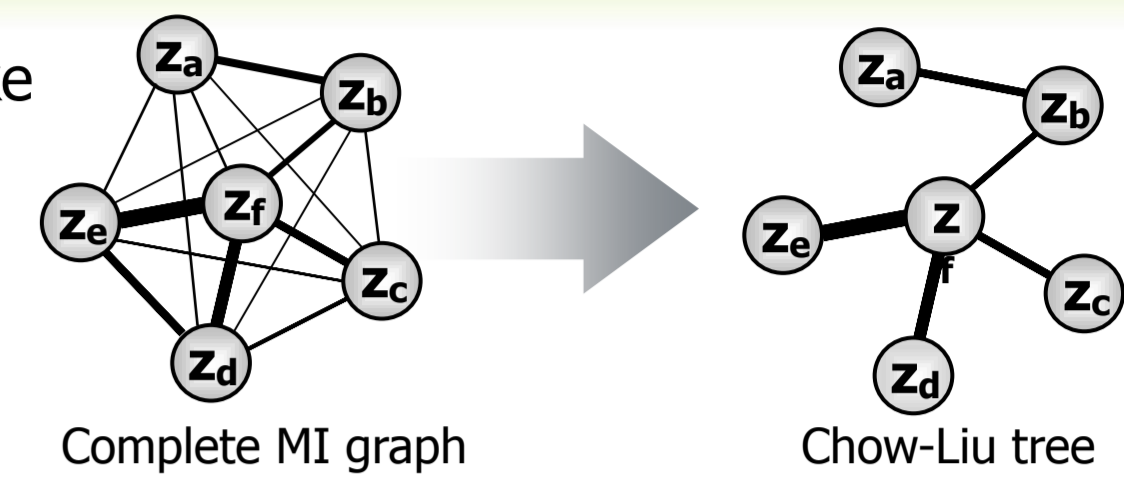
Our approach:
- Use the Chow-Liu tree as a basis for probabilistic approximations to AM.
- Relax the rigid probabilistic model of AM via approximations to the prior correlation structure preserving the dominant links.
- CLAM reduces the dense graph of probabilistic model to the Chow-Liu tree.
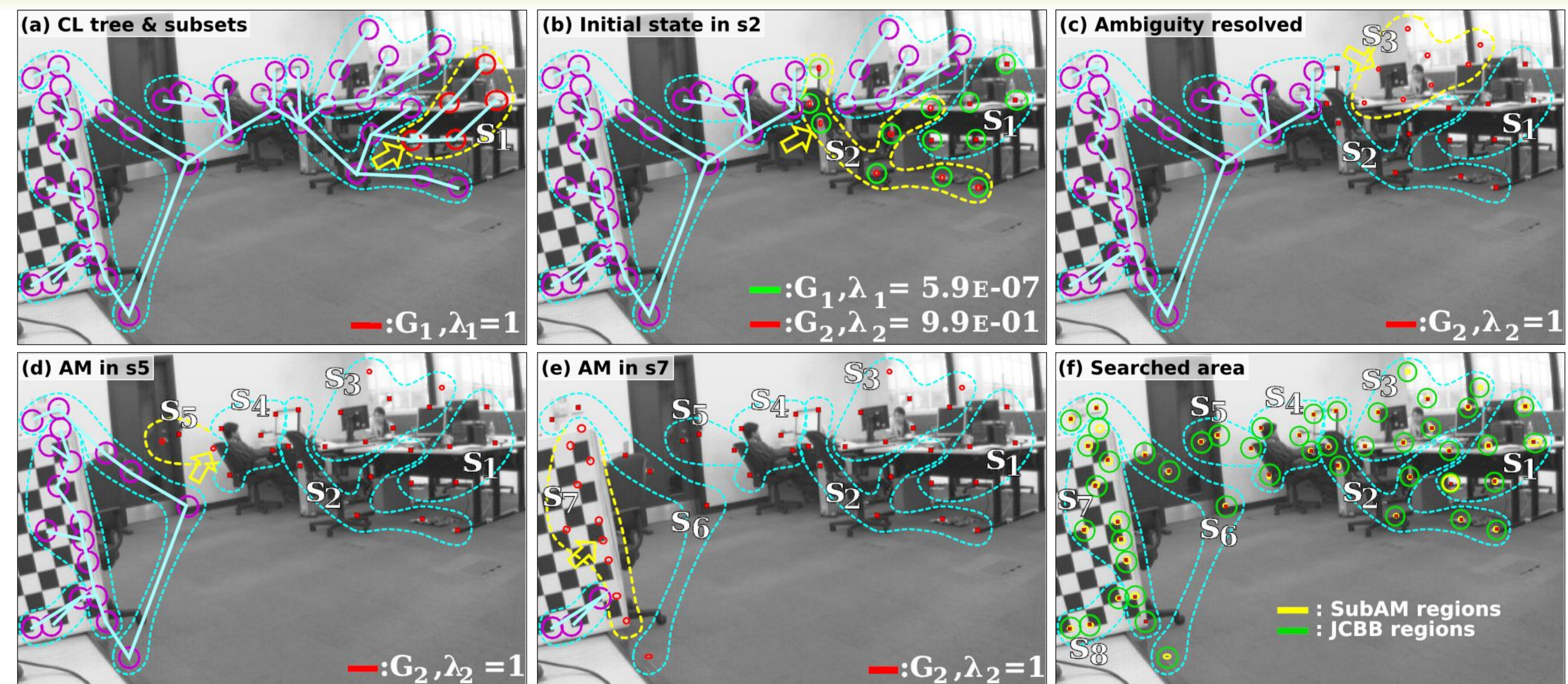- SubAM reduces the model graph to a tree of fully-connected clusters.

Prior correlation structure   CLAM: tree of nodes   SubAM: tree of clusters

4. Chow-Liu Active Matching (CLAM)

(a) Initial state: CL tree
(b) Propagate 1st measurement
(c) Failed search for a match
(d) Measure a hub-like feature
(e) Update state
(f) Searched area
: CLAM regions
: JCBB regions

- Approximates the joint PDF over feature locations with the Chow-Liu tree.
- Uses belief propagation to propagate updates to the graph following the observation of each node (feature).
- Able to efficiently match around **200 features in real-time**.

- CLAM is **efficient** because:
  - ✓ Message passing is O(n) in the worst case.
  - ✓ Matching is progressively broken down in smaller sub-trees.
- CLAM is **robust** because:
  - ✓ The Chow-Liu tree captures the most important correlation structure.
  - ✓ The approximation is conservative.

5. Subset Active Matching (SubAM)

(a) CL tree & subsets
(b) Initial state in s2
(c) Ambiguity resolved
$—:G_1,\lambda_1=1$
$—:G_1,\lambda_1= 5.9\text{E-}07$
$—:G_2,\lambda_2= 9.9\text{E-}01$
$—:G_2,\lambda_2=1$
(d) AM in s5
(e) AM in s7
(f) Searched area
$—:G_2,\lambda_2=1$
$—:G_2,\lambda_2=1$
: SubAM regions
: JCBB regions

- Approximates the joint PDF over feature locations with a tree of clusters.
- A local active matching process is run in each cluster visited in depth first order.
- Able to efficiently match **400 features in real time** (~170ms).

- SubAM is **highly efficient** because:
  - ✓ Messages are only passed to the immediate neighbouring clusters.
  - ✓ Local AM steps run in each cluster.
- SubAM is **robust** because it preserves the correlation links:
  - ✓ in each cluster.
  - ✓ between clusters in the neighbourhood.

6. Results & Conclusions

70ms for 100 features
170ms for 400 features
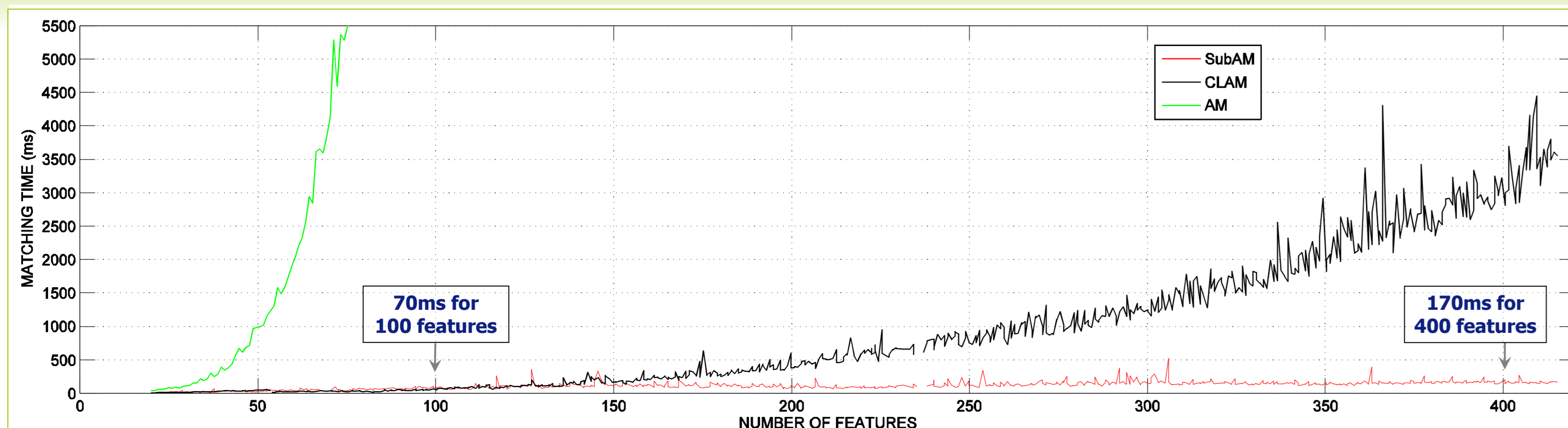
Time requirements:
- AM grows at a much faster exponential rate and is not suitable for real-time beyond 50 features.
- CLAM exhibits a vast reduction in processing time, with real-time performance feasible up to the 100-200 feature level and a relatively modest loss of speed beyond this.
- SubAM takes performance into the real-time domain for large numbers of features per frame:
  - ✓ Highly scalable
  - ✓ Can efficiently track 400 features meeting the computational demands in real-time.
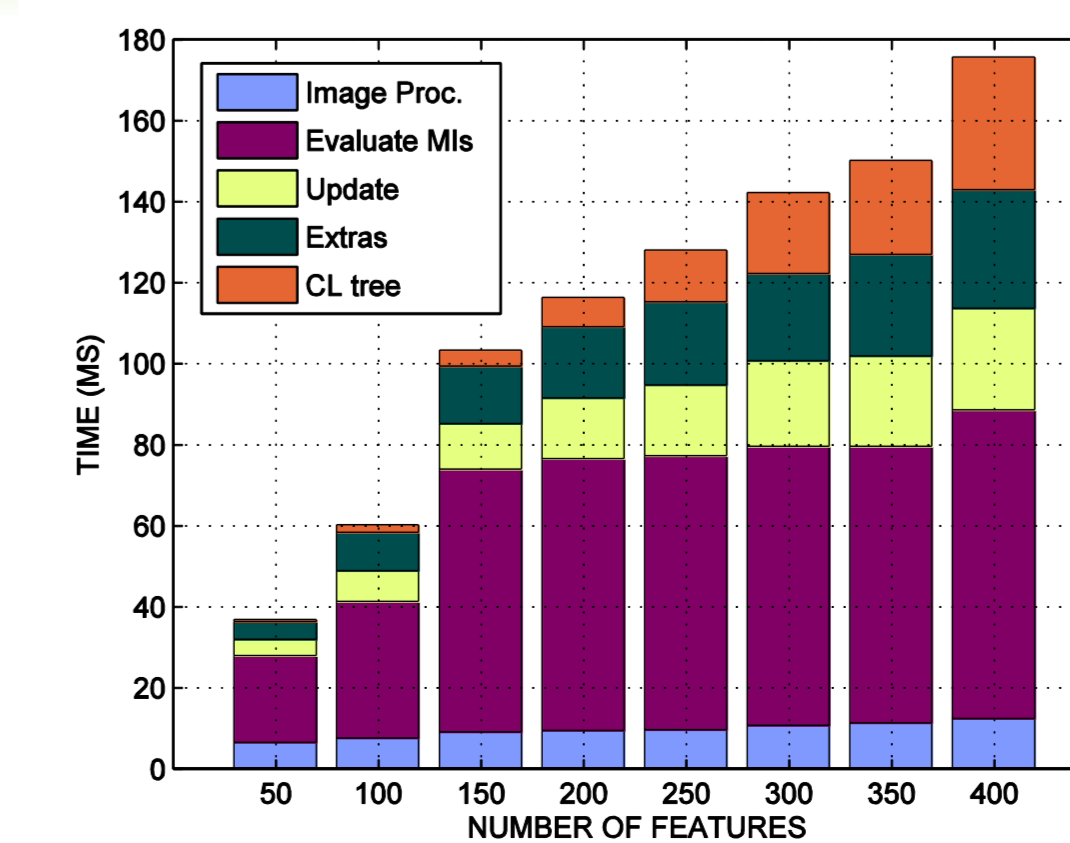
SubAM timings breakdown:
- The evaluation of MIs is the most expensive step (as is the case in AM) since AM is performed within each feature subset.
- Breaking the matching problem into highly-correlated clusters dramatically improves the scalability of fully probabilistic matching.
- Future research involves studying and employing more graph theoretic insights to do full dense matching.