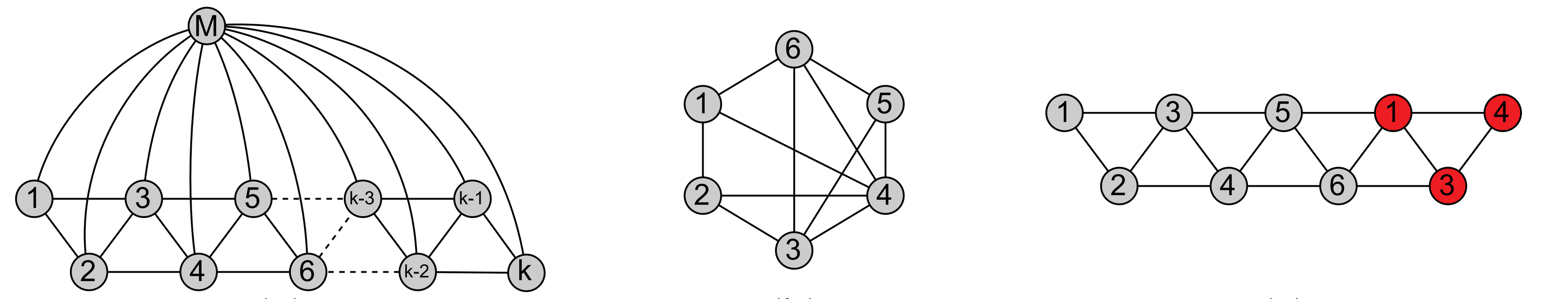
Unified Graph Matching in Euclidean Spaces

Julian McAuley, Teófilo de Campos, Tibério Caetano
julian.mcauley@nicta.com.au t.decampos@st-annes.oxon.org tiberio.caetano@nicta.com.au

Abstract

In many applications of graph matching, one is interested in finding correspondences that are **isomorphic** (i.e., the topology is preserved), as well as **isometric** (i.e., distances are preserved). While the **subgraph isomorphism** problem is known to be NP-complete, the **subgraph isometry** problem is polynomial [1]. We show that [1] can be adapted to find subgraphs that are simultaneously **both isomorphic *and* isometric** in polynomial time. Furthermore, we use **structured learning** to determine the extent to which different types of transformations are present in our model.

Our Graphical Model



(a) (b) (c)

The graphical model (a) from [2] can be used to search for *isometric* instances of the graph (b). However, it cannot identify *isomorphic* (or homeomorphic) instances, as it does not capture all of the topological constraints in (b). By replicating some of its nodes in (c), we are able to capture the missing topological constraints. The resulting model provably identifies the correct solution subject to zero noise.

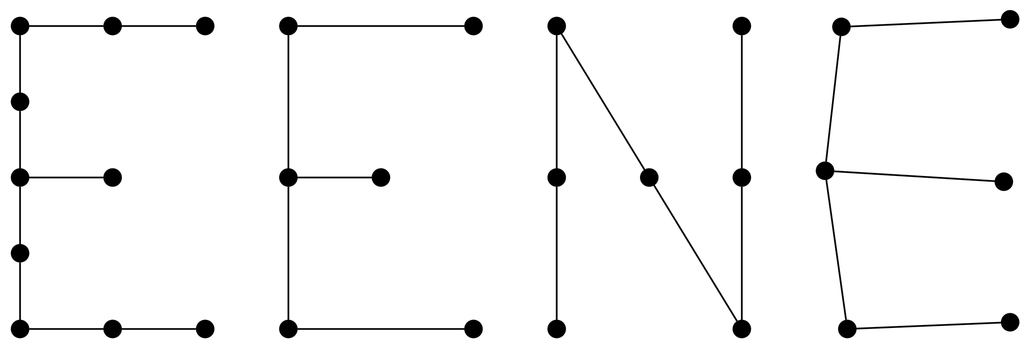
Model Parametrization

We use the structured learning approach of [4] to determine the importance of first-order features (such as Shape-Contexts and SIFT), second-order features (such as topological constraints and distances), and third-order features. Our graph-matching objective, mapping the points in V to those in V' is defined as:

$$g = \underset{f:V\to V'}{\operatorname{argmin}} \sum_{p_1\in\mathcal{G}^\sharp} \Big\langle \underbrace{\Phi^1(p_1, f(p_1))}_{\text{node features}}, \theta^{\text{nodes}} \Big\rangle + \sum_{(p_1,p_2)\in\mathcal{G}^\sharp} \Big\langle \underbrace{\Phi^2(p_1,p_2,f(p_1),f(p_2))}_{\text{edge features}}, \theta^{\text{edges}} \Big\rangle + \sum_{(p_1,p_2,p_3)\in G^\sharp} \Big\langle \underbrace{\Phi^3(p_1,p_2,p_3,f(p_1),f(p_2),f(p_3))}_{\text{triangle features}}, \theta^{\text{tri}} \Big\rangle$$

Our approach is fully-supervised, i.e., we learn $\Theta = (\theta^{\text{nodes}}; \theta^{\text{edges}}; \theta^{\text{tri}})$ from manually labeled correspondences provided by the user.

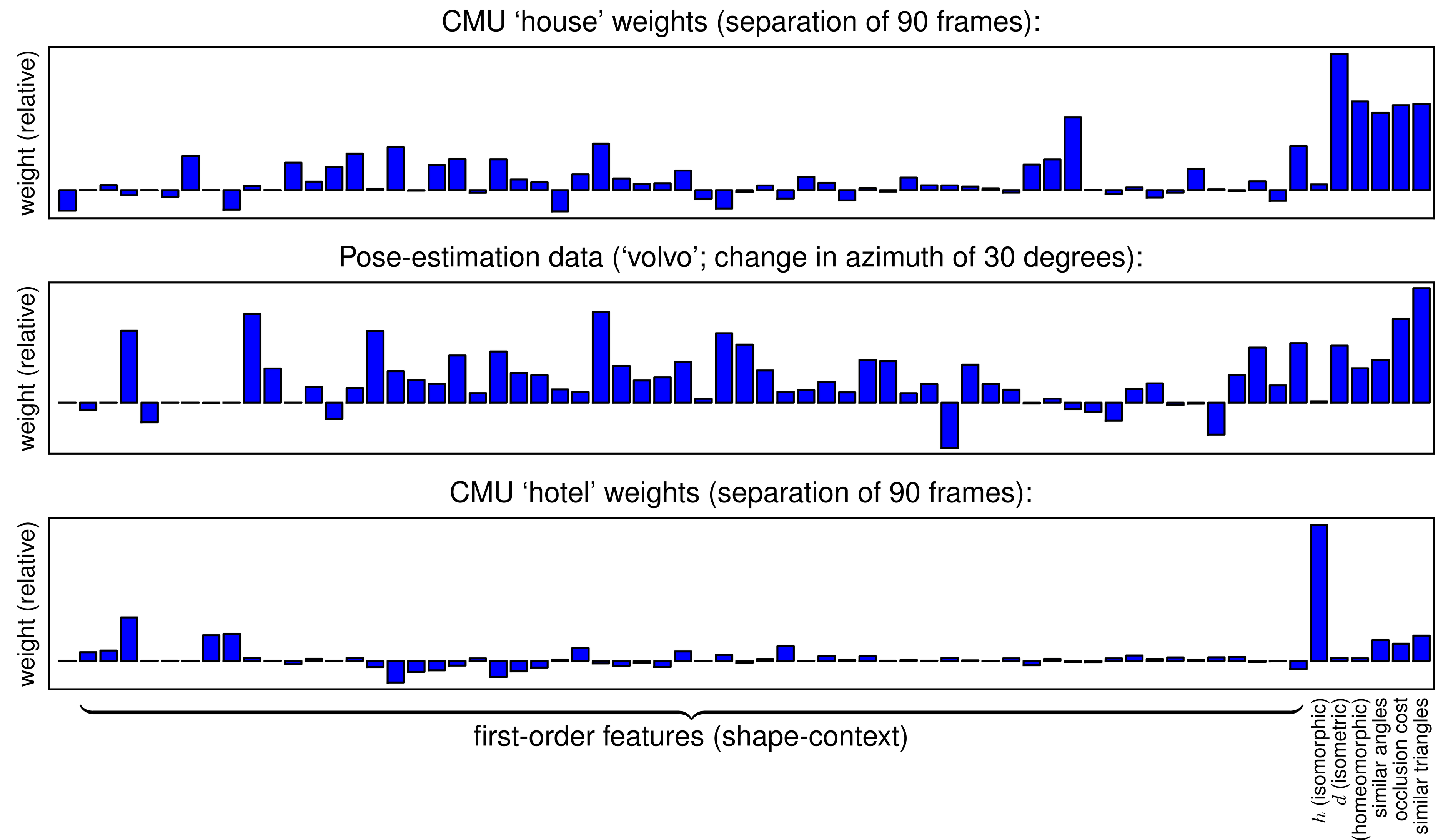
Graph Isometry and Isomorphism



(a) (b) (c) (d)

Isometry: $b \subseteq a$, $b \subseteq c$
Isomorphism: $b \subseteq d$, $c \subseteq a$, $d \subseteq b$
Homeomorphism: $b \subseteq a$, $b \subseteq d$, $c \subseteq a$, $d \subseteq a$, $d \subseteq b$

Weight Vectors



We learn a different model for each dataset. The weight vectors we learn demonstrate the varying degrees of importance of isometry, isomorphism, and homeomorphism in different applications.

bibliography

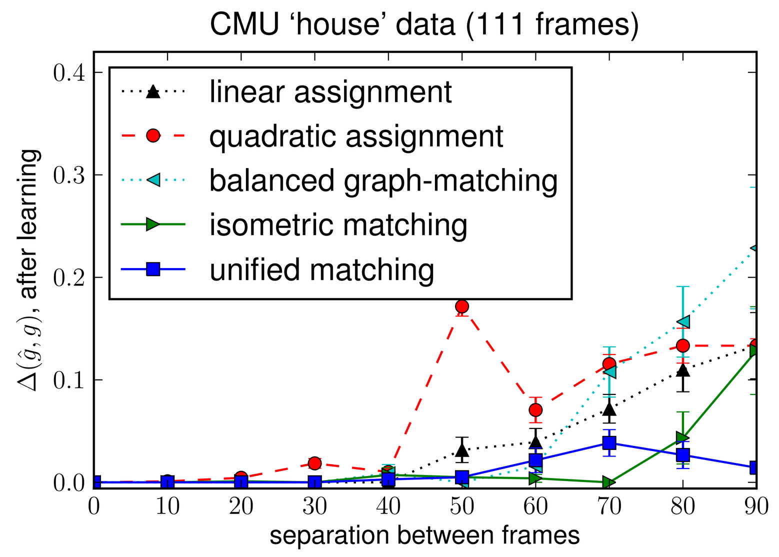
[1] T. S. Caetano and J. J. McAuley. Faster graphical models for point-pattern matching. *Spatial Vision*, 2009.

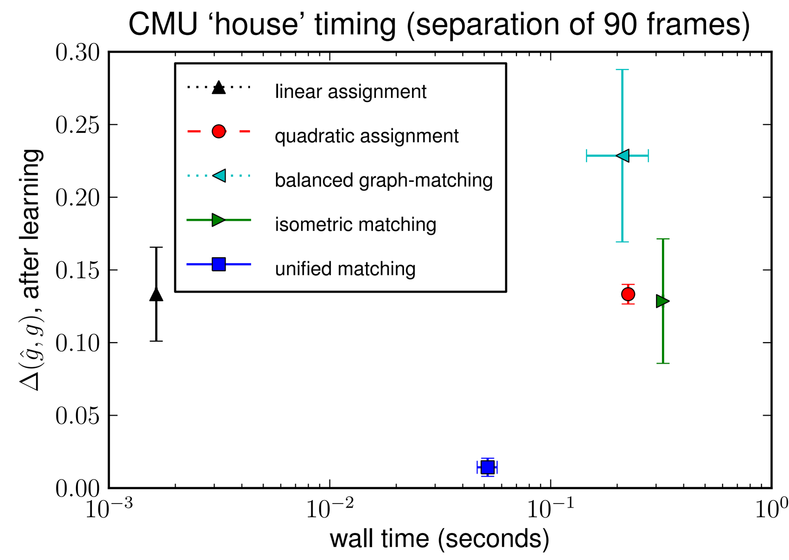
[2] J. J. McAuley, T. S. Caetano, and A. J. Smola. Robust near-isometric matching via structured learning of graphical models. *NIPS*, 2008.

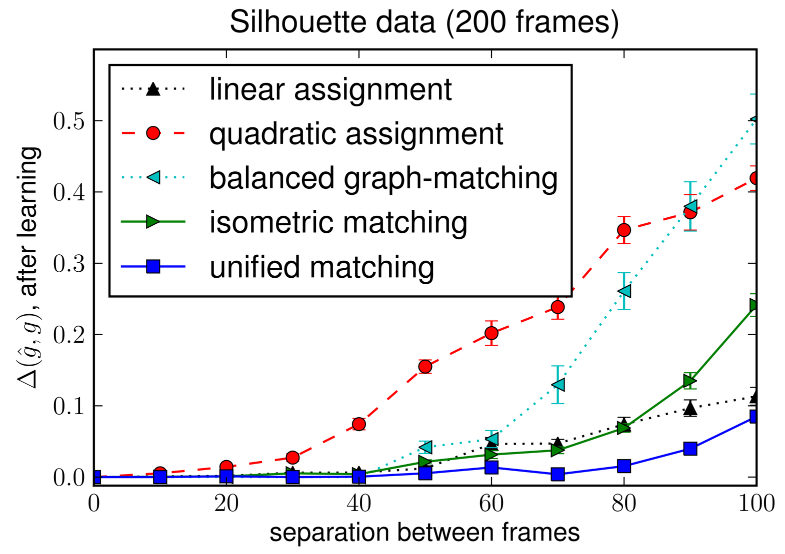
[3] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *NIPS*, 2006.

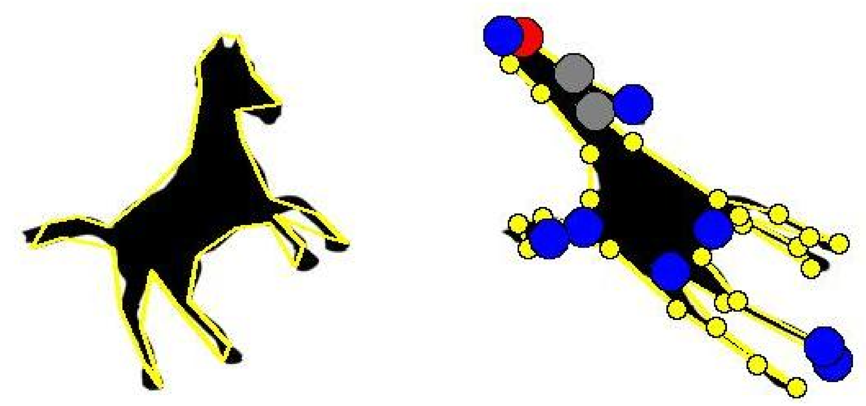
[4] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

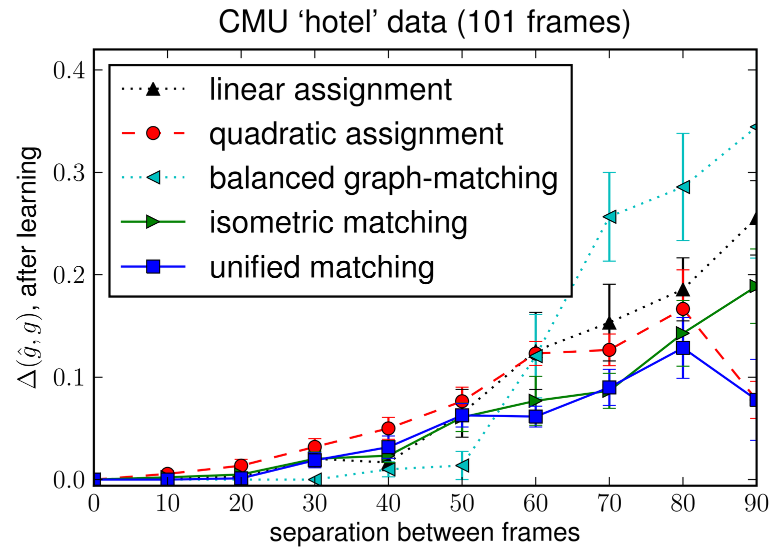
Results

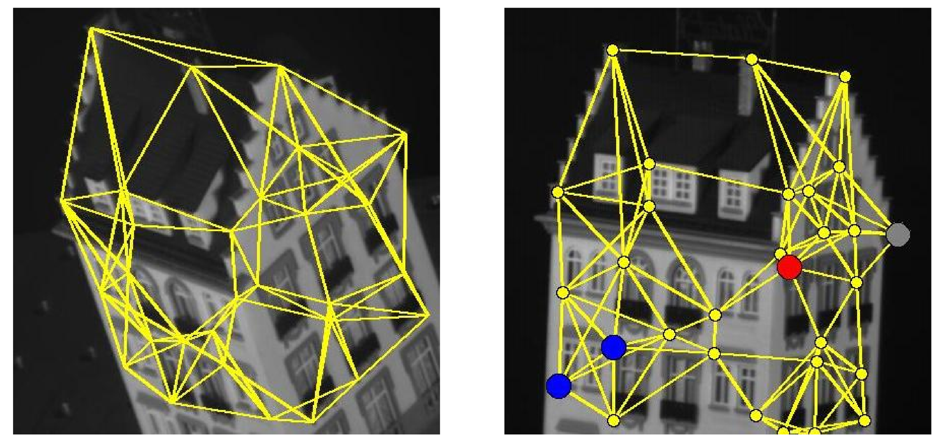


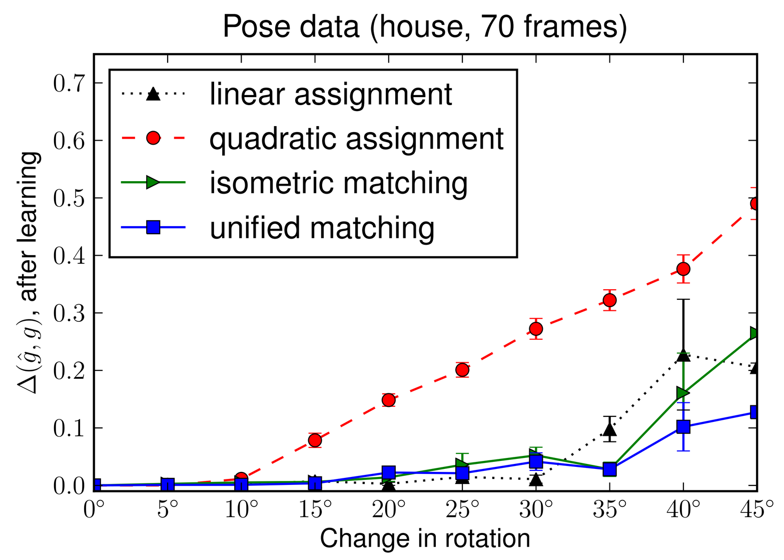


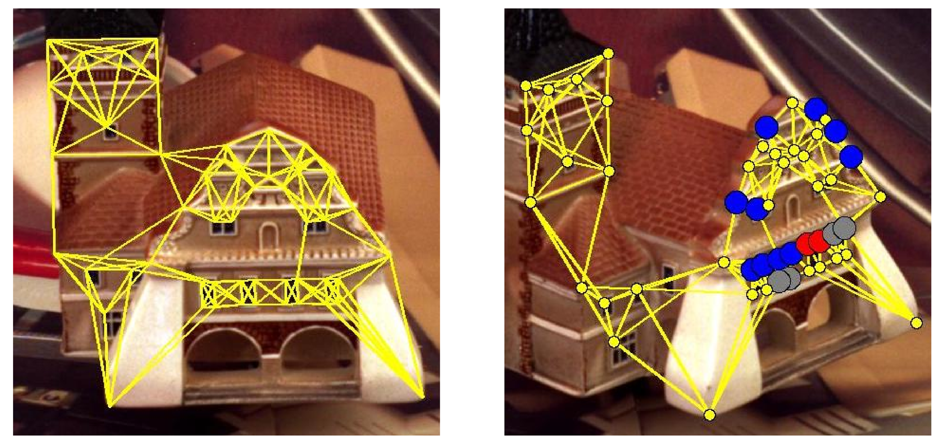


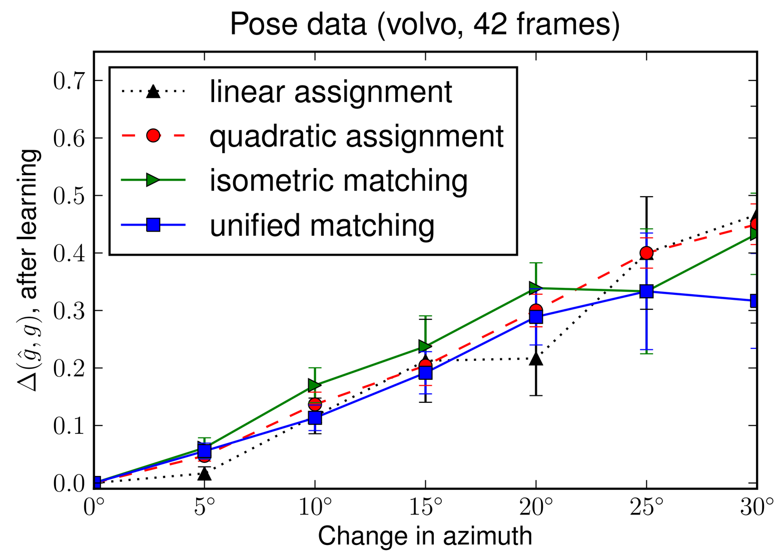


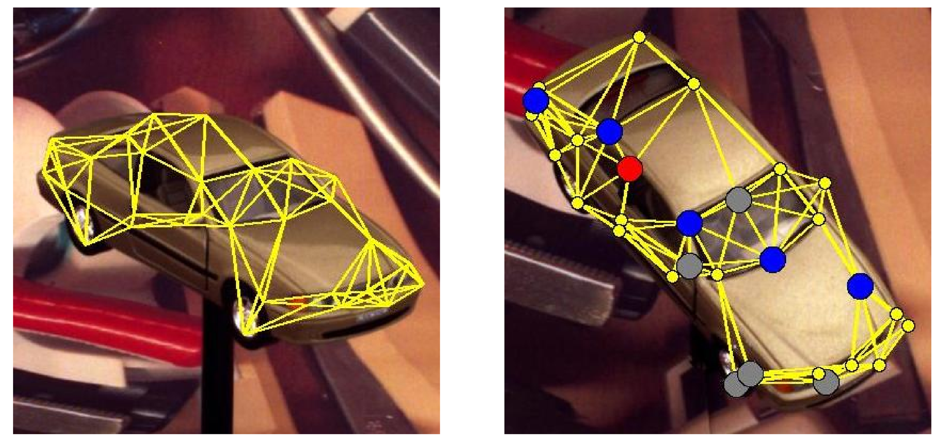












The performance of our method compared to models from [2] and [3]. The images at right compare our matching results to those from [2]; blue dots denote correspondences for which *only* our method was correct; red dots denote correspondences for which only our method was incorrect; gray dots denote that both methods were incorrect. Running times are shown at top-right.