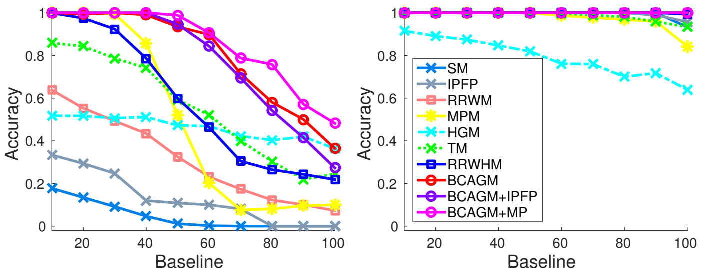
A Flexible Tensor Block Coordinate Ascent Scheme for Hypergraph Matching

Quynh Nguyen Ngoc[1,2], Antoine Gautier[2], Matthias Hein[2]
[1]Max Planck Institute for Informatics, Saarbrücken, Germany. [2]Saarland University, Saarbrücken, Germany.



(a) 10 pts vs 30 pts (b) 30 pts vs 30 pts

Figure 1: CMU house dataset: Our algorithms are denoted by BCAGM, BCAGM+IPFP, BCAGM+MP. (Best viewed in color.)

Graph resp. hypergraph matching has been used in a variety of problems in computer vision, especially in feature correspondence problems. While graph matching [1, 4] is limited to pairwise geometric relations, hypergraph matching [2, 3] can integrate better geometric information by taking into account the higher order relation between groups of points. Thus, they cope better with geometric transformation such as scaling and other forms of noise. In this paper, we present a new algorithmic framework for solving a third order hypergraph matching problem, where higher order geometric features are used.

Given two sets of feature points V and V' with $n_1 = |V| \leq n_2 = |V'|$, a third order hypergraph matching problem is formulated as

$$\max_{\mathbf{x} \in M} S^3(\mathbf{x}) := \sum_{i,j,k=1}^{n} \mathcal{F}^3_{ijk} \mathbf{x}_i \mathbf{x}_j \mathbf{x}_k. \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^{n_1 n_2}$ is a vectorized version of an assignment matrix $\mathbf{X}$ which belongs to the set $M = \{X \in \{0,1\}^{n_1 \times n_2} \mid \sum_{i=1}^{n_1} X_{ij} \leq 1, \quad \sum_{j=1}^{n_2} X_{ij} = 1\}$. By abuse of notation, we write $\mathbf{x} \in M$ to denote $\mathbf{X} \in M$. Each entry $\mathcal{F}^3_{ijk}$ of the third order symmetric tensor $\mathcal{F}^3$ roughly represents the geometric relation between three correspondences encoded in $\{i, j, k\}$. While previous work tackle this NP-hard polynomial optimization problem by relaxing the discrete constraint set M to continuous domain, we propose in this paper to directly handle the original constraint set. This is motivated by the result of previous work [1, 3, 4] which have shown the importance of one-to-one constraints in graph/hypergraph matching performance. Our main idea is to optimize instead of the original score function the multilinear form associated to it.

Multilinear Form. *The multilinear form $F^m : \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \mathbb{R}$ associated to an m-th order tensor $\mathcal{F}^m$ is given by:*

$$F^m(\mathbf{x}^1, \ldots, \mathbf{x}^m) = \sum_{i_1, \ldots, i_m}^{n} \mathcal{F}^m_{i_1 \ldots i_m} \mathbf{x}^1_{i_1} \ldots \mathbf{x}^m_{i_m},$$

and the score function $S^m : \mathbb{R}^n \to \mathbb{R}$ is defined by $S^m(\mathbf{x}) := F^m(\mathbf{x}, \ldots, \mathbf{x})$.

We show below the main steps and contributions of our paper:

1. We lift the original third order tensor $\mathcal{F}^3$ to a fourth order tensor $\mathcal{F}^4$ and show that the third and fourth order problems are equivalent,

$$\mathcal{F}^4_{ijkl} = \mathcal{F}^3_{ijk} + \mathcal{F}^3_{ijl} + \mathcal{F}^3_{ikl} + \mathcal{F}^3_{jkl}. \quad (2)$$

This lift is important as it allows us to connect the optimization of a score function with the optimization of its associated multilinear form. In particular, if S^4 is convex then it holds for all $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t} \in \mathbb{R}^n$,

(a) $F^4(\mathbf{x}, \mathbf{x}, \mathbf{y}, \mathbf{y}) \leq \max_{\mathbf{u} \in \{\mathbf{x}, \mathbf{y}\}} F^4(\mathbf{u}, \mathbf{u}, \mathbf{u}, \mathbf{u})$.

(b) $F^4(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t}) \leq \max_{\mathbf{u} \in \{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t}\}} F^4(\mathbf{u}, \mathbf{u}, \mathbf{u}, \mathbf{u})$.

2. Given S^4 is convex, we prove that the optimization of the multilinear form $F^4(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t})$ is always equivalent to the optimization of the score function $S^4(\mathbf{x})$. In particular, it holds for any compact set $D \subset \mathbb{R}^n$ that

$$\max_{\mathbf{x} \in D} F^4(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x}) = \max_{\mathbf{x}, \mathbf{y} \in D} F^4(\mathbf{x}, \mathbf{x}, \mathbf{y}, \mathbf{y}) = \max_{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t} \in D} F^4(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{t}) \quad (3)$$

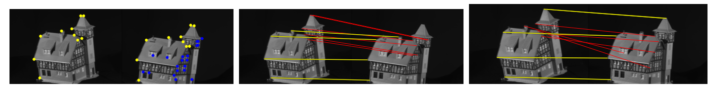
3. If S^4 is not convex, we provide a way to make it convex where the modification is constant on the set of assignment matrices M and extend this modification to a fourth order multilinear form.

4. In the algorithms, we optimize the multilinear form F^4 in a block coordinate ascent style leading to **monotonic ascent** directly on the set of assignment matrices M.

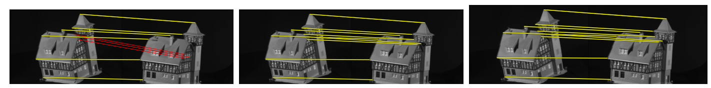
5. Our algorithms **beat all state-of-the-art methods** for third order (but also second order) graph matching in extensive experiments. Figure 1 and Figure 2 show some matching results on the CMU house dataset.

6. Our algorithms have **competitive running time** to all other methods.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.



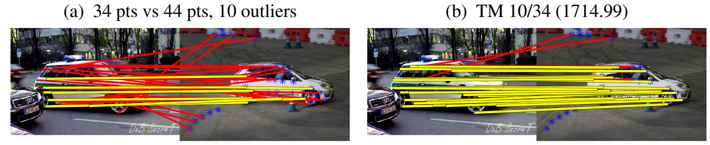
(a) Input: 10 pts vs 30 pts (b) MPM 4/10 (15.63) (c) TM 5/10 (18.37)



(d) RRWHM 7/10 (26.36) (e) BCAGM 10/10 (43.09) (f) BCAGM+MP 10/10 (43.09)

Figure 2: CMU house dataset: Blue dots indicate outlier points. Red lines indicate incorrect matches. The matching accuracy (matching score) is reported for each method. (Best viewed in color.)

(a) 34 pts vs 44 pts, 10 outliers (b) TM 10/34 (1714.99)



(c) HGM 9/34 (614.73) (d) RRWHM 28/34 (5230.52)

(e) BCAGM 28/34 (5298.77) (f) BCAGM+MP 34/34 (5377.27)

Figure 3: Car dataset: The number of correct matches and the objective score are reported. (Best viewed in color.)

[1] M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In *ECCV*, 2010.

[2] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *PAMI*, 33:2383–2395, 2011.

[3] J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted random walks. In *CVPR*, 2011.

[4] M. Leordeanu, M. Hebert, and R. Sukthankar. An integer projected fixed point method for graph matching and map inference. In *NIPS*, 2009.