A Flexible Tensor Block Coordinate Ascent Scheme for Hypergraph Matching

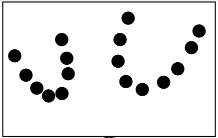
Quynh Nguyen[1,2] Antoine Gautier[2] Matthias Hein [2]

[1]Max Planck Institute for Informatics

[2]Saarland University

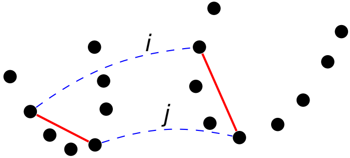
IEEE CVPR 2015, Boston, USA

Find correspondences between two sets of features / points?



Challenges: noise, deformation, geometric transformations

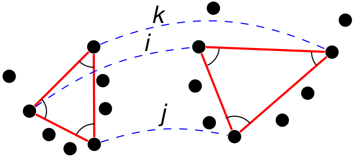
Graph Matching



- difficult to deal with rotation + scale
- Quadratic Assignment Problem

$$\max_{x \in M} \sum_{ij} \mathcal{F}_{ij} x_i x_j$$

Hypergraph Matching

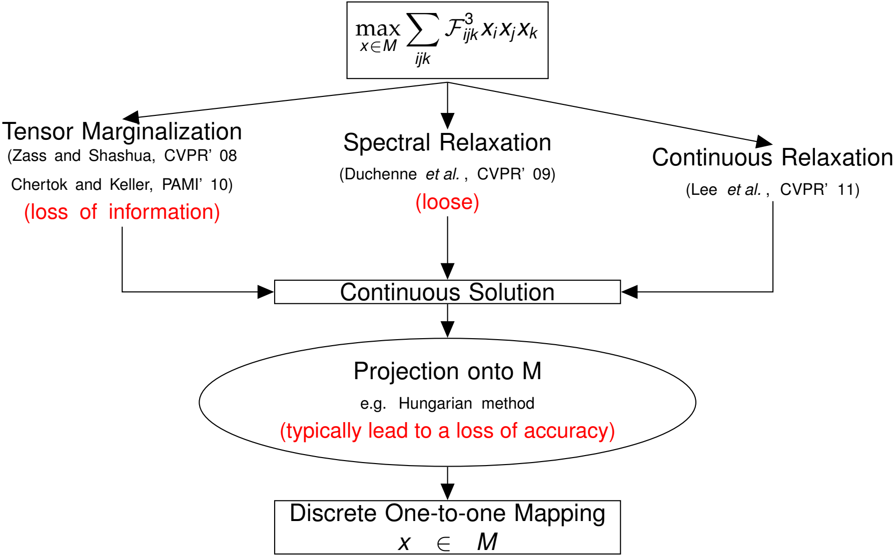


- both rotation and scale invariant
- more robust to noise
- Higher Order Assignment Problem

$$\max_{x \in M} \sum_{ijk} \mathcal{F}^3_{ijk} x_i x_j x_k$$

M = discrete one-to-one mapping constraint

M = discrete one-to-one mapping constraint



Main Concepts

- An m-dimensional array $\mathcal{F}^m$ is called an m-order tensor

  e.g. a vector is an one-dimensional tensor, a matrix is a two-dimensional tensor, etc

- One associate to any tensor $\mathcal{F}^m$ a multilinear form $F^m : \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \mathbb{R}$ such that

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

  When $x^1 = \ldots = x^m = x$, the resulting function

$$F^m(x, \ldots, x) := \sum_{i_1 \ldots i_m} \mathcal{F}^m_{i_1 \ldots i_m} x_{i_1} \ldots x_{i_m} \tag{2}$$

  is called the m-order score function

Hypergraph Matching problem becomes:

$$\max_{x \in M} \sum_{ijk} \mathcal{F}^3_{ijk} x_i x_j x_k = \boxed{\max_{x \in M} F^3(x, x, x)} \tag{3}$$

Step 1: Lifting the tensor

Given $\mathcal{F}^3 \in \mathbb{R}^{n\times n\times n}$, the lifted 4th order tensor $\mathcal{F}^4$ is defined as

$$\mathcal{F}^4_{ijkl} = \mathcal{F}^3_{ijk} + \mathcal{F}^3_{ijl} + \mathcal{F}^3_{ikl} + \mathcal{F}^3_{jkl} \quad (1)$$

$$\Longrightarrow F^4(x,x,x,x) = F^3(x,x,x)\sum_{i=1}^{n} x_i \quad (2)$$

Properties: If the new score function $F^4(x,x,x,x)$ is convex on $\mathbb{R}^n$ then

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

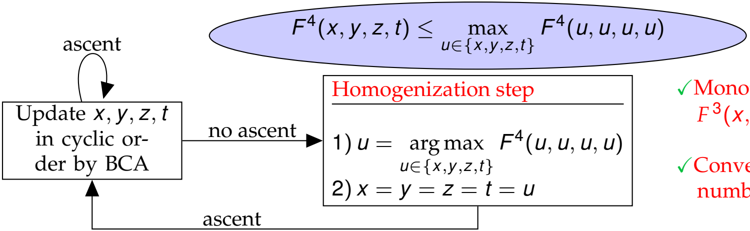
Remarks:

1. Since $\sum_i x_i = \text{constant} \quad \forall x \in M$, it holds from (2)

$$\max_{x\in M} F^4(x,x,x,x) \equiv \max_{x\in M} F^3(x,x,x)$$

2. The lifting step does not cause computational overload

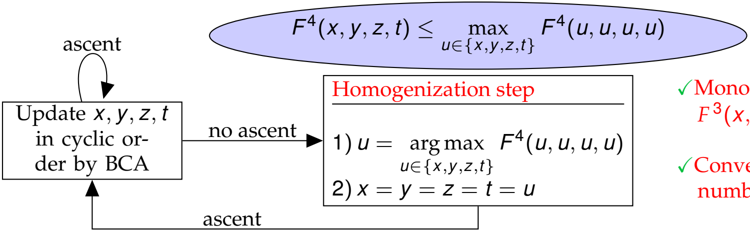
Algorithm 1: $\max_{x,y,z,t \in M} F^4(x, y, z, t)$



✓Monotonic ascent for $F^3(x,x,x)$ on M

✓Convergence in finite number of iterations

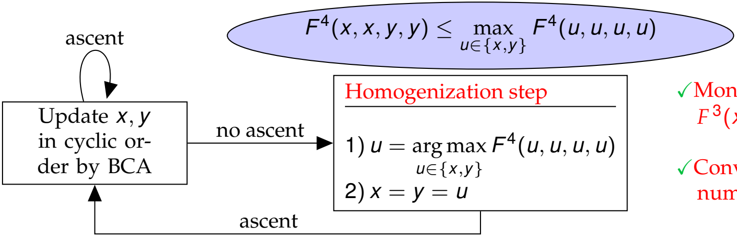
Algorithm 1: $\max_{x,y,z,t \in M} F^4(x, y, z, t)$



✓Monotonic ascent for $F^3(x, x, x)$ on M

✓Convergence in finite number of iterations

Algorithm 2: $\max_{x,y \in M} F^4(x, x, y, y)$



✓Monotonic ascent for $F^3(x, x, x)$ on M

✓Convergence in finite number of iterations

Step 2: Tackling the convexity assumption

Consider the following multilinear form:

$$F^4_\alpha(x,y,z,t) := F^4(x,y,z,t) + \alpha \frac{\langle x,y\rangle\langle z,t\rangle + \langle x,z\rangle\langle y,t\rangle + \langle x,t\rangle\langle y,z\rangle}{3} \quad (1)$$

then $F^4_\alpha(x,x,x,x)$ is convex for any $\alpha \geq 3\sqrt{\sum_{i,j,k,l=1}^{n} (\mathcal{F}^4_{ijkl})^2}$. Besides, it holds for all α

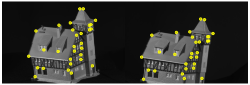
$$\max_{x\in M} F^4_\alpha(x,x,x,x) \equiv \max_{x\in M} F^3(x,x,x) \quad (2)$$

To reduce the influence of α:

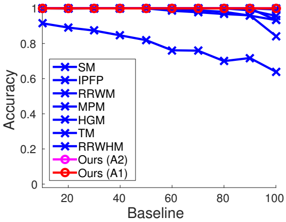
- Phase 1: run algorithm with $\alpha = 0$
- Phase 2: restart algorithm with $\alpha = 3\sqrt{\sum_{i,j,k,l=1}^{n} (\mathcal{F}^4_{ijkl})^2}$ with starting point initialized from output of Phase 1

Matching Quality (CMU House Dataset)

- 30 points are manually tracked over a sequence of frames of the same object taken at different view points
- In Figures: $\mathrm{Baseline} =$ difficulty of the problem



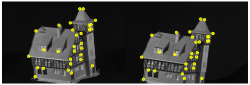
30 points vs. 30 points



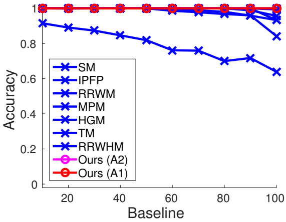
(Easy)

Matching Quality (CMU House Dataset)

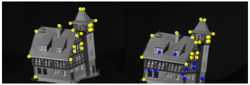
- 30 points are manually tracked over a sequence of frames of the same object taken at different view points
- In Figures: $\mathrm{Baseline} = $ difficulty of the problem



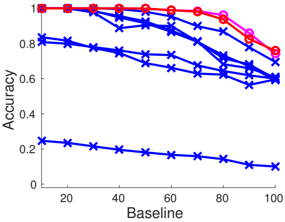
30 points vs. 30 points



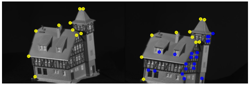
(Easy)



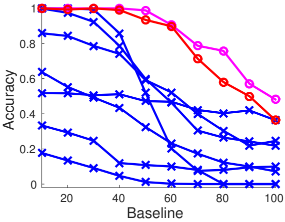
20 points vs. 30 points



(Medium)

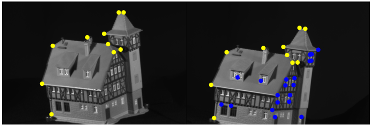


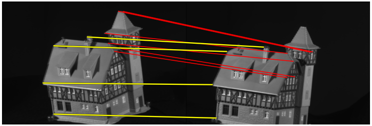
10 points vs. 30 points

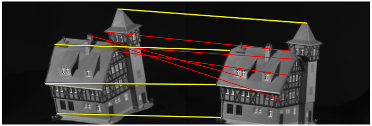


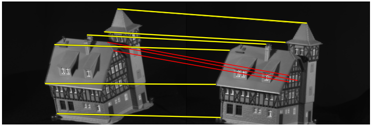
(Difficult)

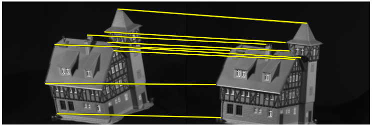
Matching Quality (CMU House Dataset)

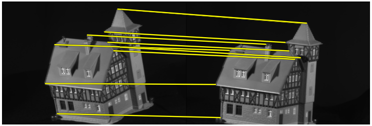

10 points vs 30 points


MPM (4/10)

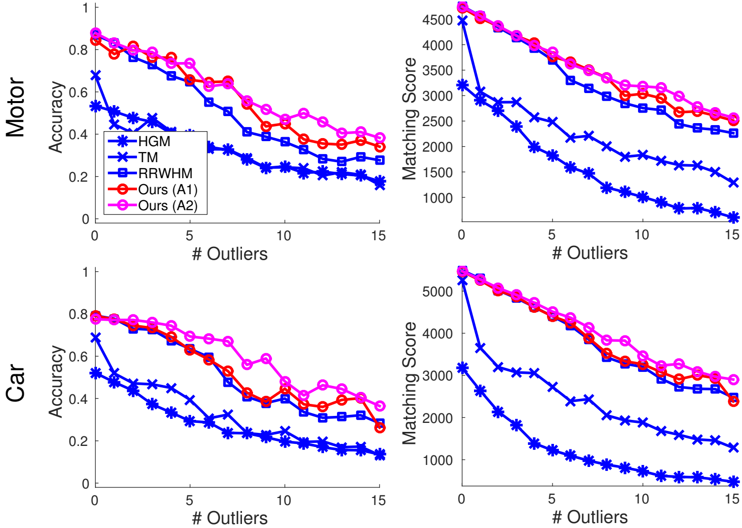

TM (5/10)


RRWHM (7/10)

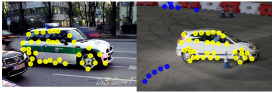

ours(A1) (10/10)


ours(A2) (10/10)

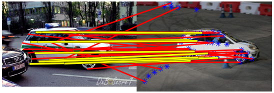
Matching Quality (Car and Motorbike Dataset)



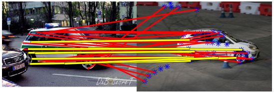
Matching Quality (Car and Motorbike Dataset)



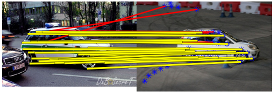
Example Input



TM (10/34)

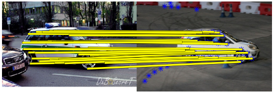


HGM (9/34)



RRWHM (28/34)

Ours(A1) (28/34)



Ours(A2) (34/34)

Conclusion

Our Contributions

- a theoretically sound optimization framework for hypergraph matching
- two state-of-the-art algorithms with theoretical guarantees
  - monotonic ascent for original score function on M
  - convergence in finite number of iterations

Main Ideas

- lifting 3rd order tensor to 4th order higher order
- multilinear optimization over one-to-one constraint

$$\max_{\mathbf{x}\in M} F^3(\mathbf{x},\mathbf{x},\mathbf{x}) \rightarrow \max_{\mathbf{x}\in M} F^4(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x}) \rightarrow \max_{\mathbf{x}\in M} F^4_\alpha(\mathbf{x},\mathbf{x},\mathbf{x},\mathbf{x})$$

$$\rightarrow \max_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{t}\in M} F^4_\alpha(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{t}) \rightarrow \text{Algorithm 1}$$

$$\rightarrow \max_{\mathbf{x},\mathbf{y}\in M} F^4_\alpha(\mathbf{x},\mathbf{x},\mathbf{y},\mathbf{y}) \rightarrow \text{Algorithm 2}$$

Advantages

- no projection at final step
- flexible objective function
- simple yet effective