Alternating Direction Graph Matching
Supplementary material

D. Khuê Lê-Huu[1, 2] Nikos Paragios[1, 2, 3]
[1]CentraleSupélec, Université Paris-Saclay [2]Inria [3]TheraPanacea
{khue.le, nikos.paragios}@centralesupelec.fr

Abstract

This is the supplementary material of [7]. We give proofs of some presented theoretical results in Appendix A and present additional experiments in Appendix B.

A. Technical Appendix

In Section A.1, we show how to obtain equation (22) in the paper. In Section A.2 we detail how ADGM1 and ADGM2 are obtained from the general decomposition framework, we show that the quadratic subproblems in the general framework are reduced to projections onto convex sets for these two algorithms. We give sketches of these algorithms in subsection A.2.3 and explain how to solve these projections in subsection A.2.4. Moreover, since the mathematical derivation of the general framework is only valid for second or higher-order graph matching, for the sake of completeness, we also show how to solve first-order graph matching (*i.e.* linear assignment) using ADGM in Section A.3, which justifies our claim that ADGM can be used to solve graph matching of *arbitrary* order. For convenience, the equation numbers in this supplement are the same as in the main paper.

A.1. Equation (22)

Denote

$$\mathbf{s}_d^k = \sum_{i=1}^{d-1} \mathbf{A}_i \mathbf{x}_i^{k+1} + \sum_{j=d+1}^{D} \mathbf{A}_j \mathbf{x}_j^k \tag{20}$$

$$\mathbf{p}_d^k = \sum_{i=d}^{D} \mathcal{F}^i \bigotimes_{j=1}^{d-1} \mathbf{x}_j^{k+1} \bigotimes_{l=d+1}^{i} \mathbf{x}_l^k. \tag{21}$$

We will prove the following result stated in the paper:

$$\sum_{i=d}^{D} F^i(\mathbf{x}_{[1,d-1]}^{k+1}, \mathbf{x}, \mathbf{x}_{[d+1,i]}^k) = (\mathbf{p}_k)^\top \mathbf{x}. \tag{22}$$

Indeed, from equation (3) in the paper, we have

$$F^i(\mathbf{x}_{[1,d-1]}^{k+1}, \mathbf{x}, \mathbf{x}_{[d+1,i]}^k) = \mathcal{F}^i \bigotimes_{j=1}^{d-1} \mathbf{x}_j^{k+1} \otimes_d \mathbf{x} \bigotimes_{l=d+1}^{i} \mathbf{x}_l^k.$$

Therefore,

$$\begin{aligned}
&\sum_{i=d}^{D} F^i(\mathbf{x}_{[1,d-1]}^{k+1}, \mathbf{x}, \mathbf{x}_{[d+1,i]}^k) \\
=&\sum_{i=d}^{D} \left(\mathcal{F}^i \bigotimes_{j=1}^{d-1} \mathbf{x}_j^{k+1} \otimes_d \mathbf{x} \bigotimes_{l=d+1}^{i} \mathbf{x}_l^k \right) \\
=&\left(\sum_{i=d}^{D} \mathcal{F}^i \bigotimes_{j=1}^{d-1} \mathbf{x}_j^{k+1} \bigotimes_{l=d+1}^{i} \mathbf{x}_l^k \right)^\top \mathbf{x} \\
=&(\mathbf{p}_k)^\top \mathbf{x}.
\end{aligned}$$

A.2. Mathematical derivation for ADGM1 and ADGM2

Recall that the $\mathbf{x}$ update step in the general decomposition problem is reduced to solving to minimizing the following quadratic functions over $\mathcal{M}_d$ ($d = 1, 2, \ldots, D$):

$$\frac{1}{2}\mathbf{x}^\top \mathbf{A}_d^\top \mathbf{A}_d \mathbf{x} + \left(\mathbf{A}_d^\top \mathbf{s}_d^k + \frac{1}{\rho}(\mathbf{A}_d^\top \mathbf{y}^k + \mathbf{p}_d^k) \right)^\top \mathbf{x}. \tag{24}$$

Recall also that $(\mathbf{A}_d)_{1\le d\le D}$ must be chosen such that the constraint $\mathbf{x}_1 = \mathbf{x}_2 = \ldots = \mathbf{x}_D$ must hold. We presented two choices in the paper, each led to a different algorithm where the subproblems (24) are reduced to

$$\mathbf{x}_d^{k+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{M}_d} \left\{ \frac{1}{2}\|\mathbf{x}\|_2^2 - \mathbf{c}_d^\top \mathbf{x} \right\}, \tag{29}$$

where $(\mathbf{c}_d)_{1\le d\le D}$ are defined by the equations (30), (31) for ADGM1 and (32), (33), (34) for ADGM2 (*c.f.* main paper). Here we show how to obtain these equations.

A.2.1 ADGM1

We choose $(\mathbf{A}_d)_{1\le d\le D}$ such that

$$\mathbf{x}_1=\mathbf{x}_2,\quad \mathbf{x}_1=\mathbf{x}_3,\dots,\quad \mathbf{x}_1=\mathbf{x}_D, \tag{27}$$

which can be re-written in matrix form:

$$\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_1\\ \vdots\\ \mathbf{x}_1\end{bmatrix}+\begin{bmatrix}-\mathbf{x}_2\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\ -\mathbf{x}_3\\ \vdots\\ \mathbf{0}\end{bmatrix}+\cdots+\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \vdots\\ -\mathbf{x}_D\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}.$$

It is straightforward that $\mathbf{A}_d$ is the d^{th} (block) **column** of the following $(D-1)\times D$ block matrix $\mathbf{A}$ where each block are $n\times n$, and as a consequence, $\mathbf{y}$ is also a $(D-1)\times 1$ block vector where each block is an n-dimensional vector:

$$\mathbf{A}=\begin{bmatrix}\mathbf{I} & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{I} & \mathbf{0} & -\mathbf{I} & \cdots & \mathbf{0}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I}\end{bmatrix},\quad \mathbf{y}=\begin{bmatrix}\mathbf{y}_2\\ \mathbf{y}_3\\ \vdots\\ \mathbf{y}_D\end{bmatrix}.$$

From (20) we easily have

$$\mathbf{s}_1^k=\begin{bmatrix}-\mathbf{x}_2^k\\ -\mathbf{x}_3^k\\ \vdots\\ -\mathbf{x}_D^k\end{bmatrix}$$

and

$$\mathbf{s}_d^k=\begin{bmatrix}\mathbf{x}_1^{k+1}\\ \vdots\\ \mathbf{x}_1^{k+1}\\ \mathbf{x}_1^{k+1}\\ \mathbf{x}_1^{k+1}\\ \vdots\\ \mathbf{x}_1^{k+1}\end{bmatrix}-\begin{bmatrix}\mathbf{x}_2^{k+1}\\ \vdots\\ \mathbf{x}_{d-1}^{k+1}\\ \mathbf{0}\\ \mathbf{x}_{d+1}^{k}\\ \vdots\\ \mathbf{x}_D^{k}\end{bmatrix}\quad \forall 2\le d\le D.$$

Now we compute the vectors $(\mathbf{c}_d)_{1\le d\le D}$.

- For $d=1$: Since $\mathbf{A}_1=\begin{bmatrix}\mathbf{I} & \mathbf{I} & \cdots & \mathbf{I}\end{bmatrix}^\top$ we have

$$\mathbf{A}_1^\top\mathbf{A}_1=(D-1)\mathbf{I},$$
$$\mathbf{A}_1^\top\mathbf{s}_1^k=-\sum_{d=2}^{D}\mathbf{x}_d^k,$$
$$\mathbf{A}_1^\top\mathbf{y}^k=\sum_{d=2}^{D}\mathbf{y}_d^k.$$

Plugging these into (24), it becomes

$$\frac{1}{2}(D-1)\left\|\mathbf{x}\right\|_2^2+\left(-\sum_{d=2}^{D}\mathbf{x}_d^k+\frac{1}{\rho}\sum_{d=2}^{D}\mathbf{y}_d^k+\frac{1}{\rho}\mathbf{p}_1^k\right)^\top\mathbf{x}.$$

Minimizing this quantity is equivalent to minimizing

$$\frac{1}{2}\left\|\mathbf{x}\right\|_2^2-\mathbf{c}_1^\top\mathbf{x}$$

where

$$\mathbf{c}_1=\frac{1}{D-1}\left(\sum_{d=2}^{D}\mathbf{x}_d^k-\frac{1}{\rho}\sum_{d=2}^{D}\mathbf{y}_d^k-\frac{1}{\rho}\mathbf{p}_1^k\right),$$

which is equation (30) in the paper.

- For $d\ge 2$: Since (the below $-\mathbf{I}$ is at the $(d-1)$-th position)

$$\mathbf{A}_d=\begin{bmatrix}\mathbf{0} & \cdots & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}^\top$$

we have

$$\mathbf{A}_d^\top\mathbf{A}_d=\mathbf{I},$$
$$\mathbf{A}_d^\top\mathbf{s}_d^k=-\mathbf{x}_1^{k+1},$$
$$\mathbf{A}_d^\top\mathbf{y}^k=-\mathbf{y}_d^k.$$

Plugging these into (24), it becomes

$$\frac{1}{2}\left\|\mathbf{x}\right\|_2^2+\left(-\mathbf{x}_1^{k+1}-\frac{1}{\rho}\mathbf{y}_d^k+\frac{1}{\rho}\mathbf{p}_d^k\right)^\top\mathbf{x}.$$

Minimizing this quantity is equivalent to minimizing

$$\frac{1}{2}\left\|\mathbf{x}\right\|_2^2-\mathbf{c}_d^\top\mathbf{x}$$

where

$$\mathbf{c}_d=\mathbf{x}_1^{k+1}+\frac{1}{\rho}\mathbf{y}_d^k-\frac{1}{\rho}\mathbf{p}_d^k,$$

which is equation (31) in the paper.

A.2.2 ADGM2

We choose $(\mathbf{A}_d)_{1\le d\le D}$ such that

$$\mathbf{x}_1=\mathbf{x}_2,\quad \mathbf{x}_2=\mathbf{x}_3,\dots,\quad \mathbf{x}_{D-1}=\mathbf{x}_D, \tag{28}$$

which can be re-written in matrix form:

$$\begin{bmatrix}\mathbf{x}_1\\ \mathbf{0}\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}+\begin{bmatrix}-\mathbf{x}_2\\ \mathbf{x}_2\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}+\begin{bmatrix}\mathbf{0}\\ -\mathbf{x}_3\\ \mathbf{x}_3\\ \vdots\\ \mathbf{0}\end{bmatrix}+\cdots+\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\\ -\mathbf{x}_D\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}.$$

Similarly to ADGM1, $\mathbf{A}_d$ is chosen to be the d^{th} (block) **column** of the following $(D-1)\times D$ block matrix $\mathbf{A}$ and

$\mathbf{y}$ is also a $(D-1)\times 1$ block vector:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & -\mathbf{I} & & & & \\ & \mathbf{I} & -\mathbf{I} & & & \\ & & \mathbf{I} & \ddots & & \\ & & & \ddots & -\mathbf{I} & \\ & & & & \mathbf{I} & -\mathbf{I} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_D \end{bmatrix}. \tag{29}$$

From (20) we easily have

$$\mathbf{s}_1^k = \begin{bmatrix} \mathbf{0} - \mathbf{x}_2^k \\ \mathbf{x}_2^k - \mathbf{x}_3^k \\ \mathbf{x}_3^k - \mathbf{x}_4^k \\ \vdots \\ \mathbf{x}_{D-1}^k - \mathbf{x}_D^k \end{bmatrix}, \quad \mathbf{s}_D^k = \begin{bmatrix} \mathbf{x}_1^{k+1} - \mathbf{x}_2^{k+1} \\ \mathbf{x}_2^{k+1} - \mathbf{x}_2^{k+1} \\ \vdots \\ \mathbf{x}_{D-2}^{k+1} - \mathbf{x}_{D-1}^{k+1} \\ \mathbf{x}_{D-1}^{k+1} - \mathbf{0} \end{bmatrix}$$

and

$$\mathbf{s}_d^k = \begin{bmatrix} \mathbf{x}_1^{k+1} - \mathbf{x}_2^{k+1} \\ \vdots \\ \mathbf{x}_{d-2}^{k+1} - \mathbf{x}_{d-1}^{k+1} \\ \mathbf{x}_{d-1}^{k+1} - \mathbf{0} \\ \mathbf{0} - \mathbf{x}_{d+1}^k \\ \mathbf{x}_{d+1}^k - \mathbf{x}_{d+2}^k \\ \vdots \\ \mathbf{x}_{D-1}^k - \mathbf{x}_D^k \end{bmatrix} \quad \forall 2 \leq d \leq D-1.$$

Now we compute the vectors $(\mathbf{c}_d)_{1\leq d\leq D}$.

- For $d=1$: Since $\mathbf{A}_1 = \begin{bmatrix}\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}^\top$ we have

$$\begin{aligned} \mathbf{A}_1^\top \mathbf{A}_1 &= \mathbf{I}, \\ \mathbf{A}_1^\top \mathbf{s}_1^k &= -\mathbf{x}_2^k, \\ \mathbf{A}_1^\top \mathbf{y}^k &= \mathbf{y}_2^k. \end{aligned}$$

  Plugging these into (24), it becomes

$$\frac{1}{2}\|\mathbf{x}\|_2^2 + \left(-\mathbf{x}_2^k + \frac{1}{\rho}\mathbf{y}_2^k + \frac{1}{\rho}\mathbf{p}_1^k\right)^\top \mathbf{x}.$$

  Minimizing this quantity is equivalent to minimizing

$$\frac{1}{2}\|\mathbf{x}\|_2^2 - \mathbf{c}_1^\top \mathbf{x}$$

  where

$$\mathbf{c}_1 = \mathbf{x}_2^k - \frac{1}{\rho}\mathbf{y}_2^k - \frac{1}{\rho}\mathbf{p}_1^k,$$

  which is equation (32) in the paper.

- For $d=D$: Since $\mathbf{A}_D = \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{0} & -\mathbf{I}\end{bmatrix}^\top$ we have

$$\begin{aligned} \mathbf{A}_D^\top \mathbf{A}_D &= \mathbf{I}, \\ \mathbf{A}_D^\top \mathbf{s}_D^k &= -\mathbf{x}_{D-1}^{k+1}, \\ \mathbf{A}_D^\top \mathbf{y}^k &= -\mathbf{y}_D^k. \end{aligned}$$

  Plugging these into (24), it becomes

$$\frac{1}{2}\|\mathbf{x}\|_2^2 + \left(-\mathbf{x}_{D-1}^k - \frac{1}{\rho}\mathbf{y}_D^k + \frac{1}{\rho}\mathbf{p}_D^k\right)^\top \mathbf{x}.$$

  Minimizing this quantity is equivalent to minimizing

$$\frac{1}{2}\|\mathbf{x}\|_2^2 - \mathbf{c}_D^\top \mathbf{x}$$

  where

$$\mathbf{c}_D = \mathbf{x}_{D-1}^{k+1} + \frac{1}{\rho}\mathbf{y}_D^k - \frac{1}{\rho}\mathbf{p}_D^k,$$

  which is equation (33) in the paper.

- For $2 \leq d \leq D-1$: Since (the below non-zero blocks are at the $(d-1)$-th and d-th positions)

$$\mathbf{A}_d = \begin{bmatrix}\mathbf{0} & \cdots & \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}^\top$$

  we have

$$\begin{aligned} \mathbf{A}_d^\top \mathbf{A}_d &= 2\mathbf{I}, \\ \mathbf{A}_d^\top \mathbf{s}_d^k &= -\mathbf{x}_{d-1}^{k+1} - \mathbf{x}_{d+1}^k, \\ \mathbf{A}_d^\top \mathbf{y}^k &= -\mathbf{y}_d^k + \mathbf{y}_{d+1}^k. \end{aligned}$$

  Plugging these into (24), it becomes

$$\|\mathbf{x}\|_2^2 + \left(-\mathbf{x}_{d-1}^{k+1} - \mathbf{x}_{d+1}^k - \frac{1}{\rho}(\mathbf{y}_d^k - \mathbf{y}_{d+1}^k) + \frac{1}{\rho}\mathbf{p}_d^k\right)^\top \mathbf{x}.$$

  Minimizing this quantity is equivalent to minimizing

$$\frac{1}{2}\|\mathbf{x}\|_2^2 - \mathbf{c}_d^\top \mathbf{x}$$

  where

$$\mathbf{c}_d = \frac{1}{2}(\mathbf{x}_{d-1}^{k+1} + \mathbf{x}_{d+1}^k) + \frac{1}{2\rho}(\mathbf{y}_d^k - \mathbf{y}_{d+1}^k) - \frac{1}{2\rho}\mathbf{p}_d^k,$$

  which is equation (34) in the paper.

A.2.3 Sketches of algorithms

The above algorithms are summarized in Table 1.

A.2.4 Solving the subproblems

We have seen how the subproblems in the two above ADGM algorithms can be reduced to

$$\mathbf{x}_d^{k+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{M}_d} \left\{\frac{1}{2}\|\mathbf{x}\|_2^2 - \mathbf{c}_d^\top \mathbf{x}\right\},$$

which means that $\mathbf{x}_d^{k+1}$ is the projection of $\mathbf{c}_d$ onto $\mathcal{M}_d$. Recall that $\mathcal{M}_d$ is equal to either $\mathcal{M}_\mathrm{r}$ or $\mathcal{M}_\mathrm{c}$, defined by (25) and (26) in the paper, *i.e.* either the sum of each row of $\mathbf{x}$ is ≤ 1, or the sum of each column of $\mathbf{x}$ is ≤ 1 (if no occlusion is allowed then "≤ 1 is" is replaced by "$= 1$"). Therefore, the projection is reduced to projection of each row or column of $\mathbf{x}$, which can be solved using the following lemma.

ADGM1 – ADGM2
0. Input: $\rho, N, \epsilon, (\mathbf{x}_d^0)_{1\leq d\leq D}, (\mathbf{y}_d^0)_{2\leq d\leq D}$.   1. Repeat for $k = 0, 1, 2, \ldots$:   (a) For $d = 1, 2, \ldots, D$: $$\mathbf{x}_d^{k+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathcal{M}_d} \left\{ \frac{1}{2}\|\mathbf{x}\|_2^2 - \mathbf{c}_d^\top \mathbf{x} \right\},$$ where $(\mathbf{c}_d)_{1\leq d\leq D}$ are defined by (30), (31) for ADGM1 and by (32), (33), (34) for ADGM2 (*c.f.* main paper for the equation numbers).   (b) For $d = 2, 3, \ldots, D$:   ADGM1: $\mathbf{y}_d^{k+1} = \mathbf{y}_d^k + \rho(\mathbf{x}_1^{k+1} - \mathbf{x}_d^{k+1})$.   ADGM2: $\mathbf{y}_d^{k+1} = \mathbf{y}_d^k + \rho(\mathbf{x}_{d-1}^{k+1} - \mathbf{x}_d^{k+1})$.   (c) Compute the residual r^{k+1}.   Stop if $r^{k+1} \leq \epsilon$ or $k \geq N$.   2. Discretize $\mathbf{x}_1$ and return.

Table 1: Examples of ADGM algorithms for solving D^{th}-order graph matching.

Lemma 1 *Let d be a positive integer and $\mathbf{c} = (c_1, c_2, \ldots, c_d)$ be a real-valued constant vector. Consider the problem of minimizing*

$$\frac{1}{2}\|\mathbf{u}\|_2^2 - \mathbf{c}^\top \mathbf{u} \tag{A.1}$$

with respect to $\mathbf{u} \in \mathbb{R}^d$, subject to one of the following sets of constraints:

(a) $\mathbf{u} \geq \mathbf{0}$ *and* $\mathbf{1}^\top \mathbf{u} = 1$, i.e. $\mathbf{u}$ *belongs to the* probability simplex.

(b) $\mathbf{u} \geq \mathbf{0}$ *and* $\mathbf{1}^\top \mathbf{u} \leq 1$, i.e. $\mathbf{u}$ *belongs to the* unit simplex.

An optimal solution $\mathbf{u}^*$ to each of the above two cases is given as follows:

(a) Let $\mathbf{a} = (a_1, a_2, \ldots, a_d)$ be a decreasing permutation of $\mathbf{c}$ via a permutation function σ, i.e. $a_i = c_{\sigma(i)}$*and $a_1 \geq a_2 \geq \cdots \geq a_d$. Denote*

$$\lambda_k = \frac{1}{k}\left(\sum_{1\leq i\leq k} a_i - 1\right) \quad \forall k \in \mathbb{Z}, 1 \leq k \leq d. \tag{A.2}$$

Then there exists $k^* \in \mathbb{Z}, 1 \leq k^* \leq d$, such that $a_{k^*} > \lambda_{k^*} \geq a_{k^*+1}$. An optimal solution $\mathbf{u}^* = (u_1^*, u_2^*, \ldots, u_d^*)$ is given by:

$$u_{\sigma(i)}^* = \begin{cases} a_i - \lambda_{k^*} & \text{if } 1 \leq i \leq k^*, \\ 0 & \text{if } k^* < i \leq d. \end{cases} \tag{A.3}$$

(b) Let $\mathbf{u}_0 = \max(\mathbf{c}, \mathbf{0})$. We have:

- *If $\mathbf{1}^\top \mathbf{u}_0 \leq 1$ then $\mathbf{u}^* = \mathbf{u}_0$.*
- *Otherwise, any optimal solution $\mathbf{u}^*$ must satisfy $\mathbf{1}^\top \mathbf{u}^* = 1$. Thus, the problem is reduced to the previous case, and as a consequence, $\mathbf{u}^*$ is given by* (A.3).

Proof. For part (a), see for example [4]. For part (b), the corresponding KKT conditions are:

$$\mathbf{u} \geq \mathbf{0}, \tag{A.4}$$
$$\mathbf{1}^\top \mathbf{u} \leq 1, \tag{A.5}$$
$$\boldsymbol{\mu} \geq \mathbf{0}, \tag{A.6}$$
$$\nu \geq 0, \tag{A.7}$$
$$\mu_i u_i = 0 \quad \forall 1 \leq i \leq d, \tag{A.8}$$
$$\nu(\mathbf{1}^\top \mathbf{u} - 1) = 0, \tag{A.9}$$
$$u_i - c_i + \nu - \mu_i = 0 \quad \forall 1 \leq i \leq d. \tag{A.10}$$

If $\nu = 0$ then from (A.4), (A.6), (A.8) and (A.10) we have

$$u_i \geq 0 \quad \forall i, \tag{A.11}$$
$$u_i - c_i = \mu_i \geq 0 \quad \forall i, \tag{A.12}$$
$$u_i(u_i - c_i) = 0 \quad \forall i, \tag{A.13}$$

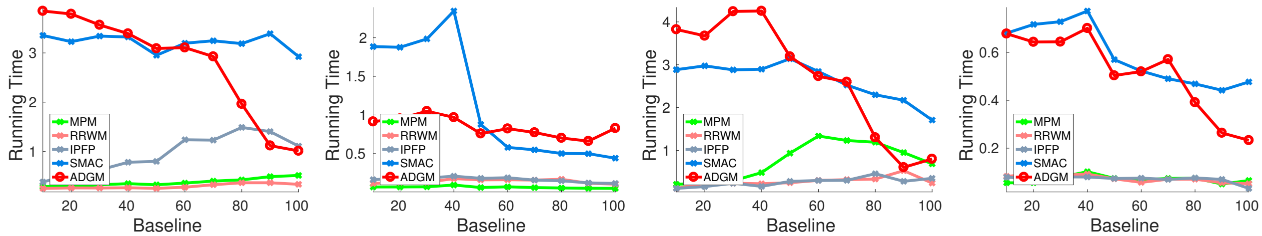
which yields $\mathbf{u} = \mathbf{u}_0$ where $\mathbf{u}_0 = \max(\mathbf{c}, \mathbf{0})$. Thus, if $\mathbf{1}^\top \mathbf{u} \leq 1$ then $\mathbf{u}_0$ is the optimal solution. Otherwise, ν must be different from 0. In this case, from (A.9), any optimal solution must satisfy $\mathbf{1}^\top \mathbf{u} = 1$ and thus, the problem is reduced to part (a). □

In our implementation, we used the fast projection algorithm introduced in [4].

A.3. First-order ADGM

Recall that ADGM1 imposes $\mathbf{x}_1 = \mathbf{x}_d \quad \forall\, 2 \leq d \leq D$ and ADGM2 imposes $\mathbf{x}_{d-1} = \mathbf{x}_d \quad \forall\, 2 \leq d \leq D$. Clearly, these constraints are only valid for $D \geq 2$ and when $D = 2$ these two sets of constraints become the same, *i.e.* ADGM1 and ADGM2 are identical.

For the sake of completeness, we briefly consider the case $D = 1$, *i.e.* first-order graph matching (also called the *linear assignment problem*). This problem can be seen as a special case of pairwise graph matching where the pairwise potentials are zeros. The problem can be reformulated as minimizing $F^1(\mathbf{x}_1)$ subject to $\mathbf{x}_1 = \mathbf{x}_2$ and $\mathbf{x}_1 \in \mathcal{M}_1, \mathbf{x}_2 \in \mathcal{M}_2$ (we can choose $\mathcal{M}_1 = \mathcal{M}_\text{r}$ and $\mathcal{M}_2 = \mathcal{M}_\text{c}$). Since the objective function is convex and separable, ADGM is guaranteed to produce a global optimum (see *e.g.* [1]) to the continuous relaxation of the matching. However, it is well-known that this continuous relaxation is just equivalent to the original problem [11, Chapter 18]. Therefore, ADGM also produces a global optimum to the linear assignment problem.



(a) House: 20 pts vs 30 pts (b) House: 10 pts vs 30 pts (c) Hotel: 20 pts vs 30 pts (d) Hotel: 10 pts vs 30 pts

Figure 1: Running time on House and Hotel sequences using the **pairwise model B** described in Section 4.1.

B. Experimental Appendix

In this appendix we report the running time of each algorithm and present additional results. ADGM algorithms are compared with the following methods:

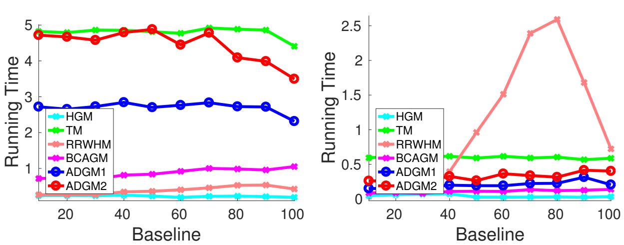
Pairwise: Spectral Matching with Affine Constraint (SMAC) [5], Integer Projected Fixed Point (IPFP) [9], Reweighted Random Walk Matching (RRWM) [2] and Max-Pooling Matching (MPM) [3].

Higher-order: Probabilistic Graph Matching (HGM) [12], Tensor Matching (TM) [6], Reweighted Random Walk Hypergraph Matching (RRWHM) [8] and Block Coordinate Ascent Graph Matching (BCAGM) [10].

B.1. House and Hotel

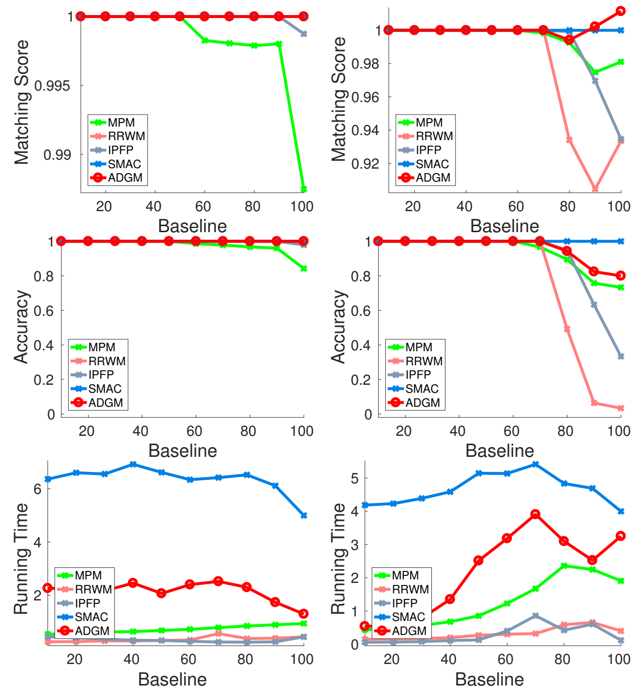
Pairwise model B. We have presented the results of matching 20 points to 30 points and 10 points to 30 points in the paper. Running time results for these experiments are given in Figure 1. Results of matching 30 points to 30 points are given in Figure 3.

Third-order model. We have presented the results of matching 20 points to 30 points and 10 points to 30 points on the **House** sequence using the **third-order model** described in Section 4.1 in the paper. The running time results for these experiments are given in Figure 2. The results of the other cases are given in Figure 4.



(a) House: 20 pts vs 30 pts (b) House: 10 pts vs 30 pts

Figure 2: Running time on **House** sequence using the **third-order model** described in Section 4.1.



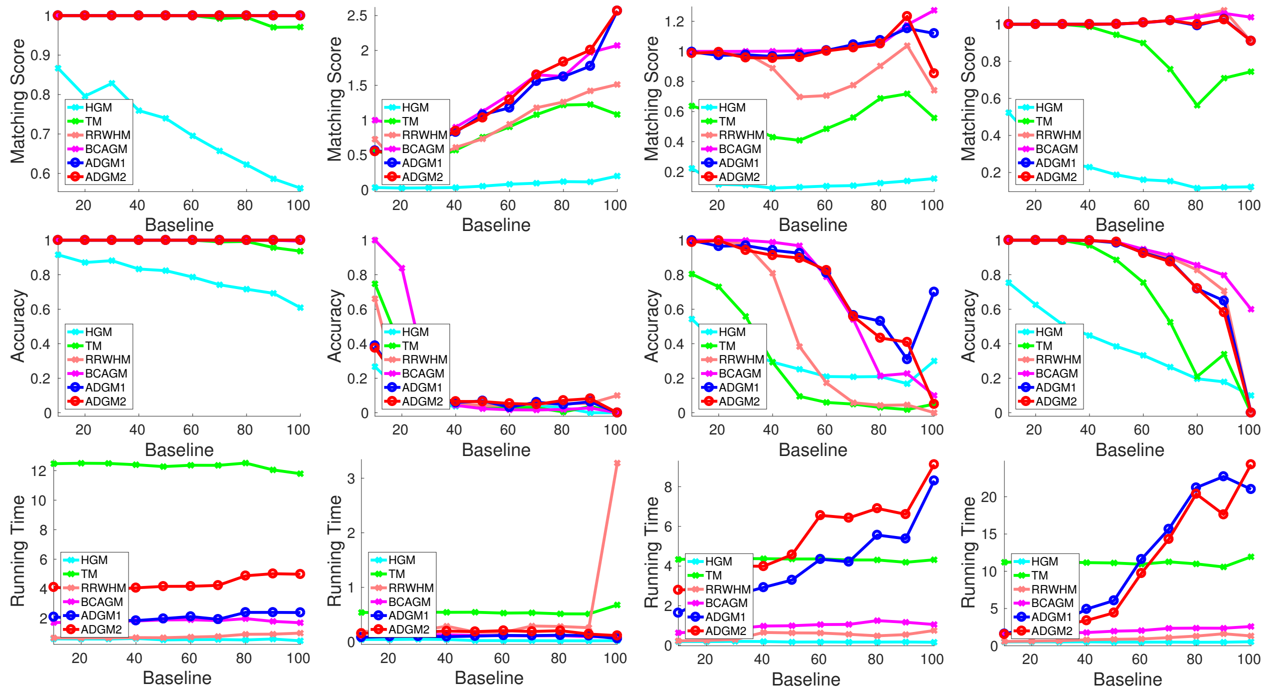
(a) House: 30 pts vs 30 pts (b) Hotel: 30 pts vs 30 pts

Figure 3: Matching 30 points to 30 points on House and Hotel sequences using the **pairwise model B** described in Section 4.1.

B.2. Cars and Motorbikes

Pairwise model B. We have stated in the paper that using this model (described in Section 4.1 of the paper), the obtained results are unsatisfactory. Indeed, one can observe from the results in Figure 5 that the obtained accuracies are very low, even though ADGM always achieved the best objective values that are higher than the ground-truth ones. One can conclude that this pairwise model is not suited for this dataset.

Pairwise model C. This model has been shown to be very suited for this dataset of real images. The running time results of the experiments presented in the paper are given in



(a) House: 30 pts vs 30 pts (b) Hotel: 10 pts vs 30 pts (c) Hotel: 20 pts vs 30 pts (d) Hotel: 30 pts vs 30 pts

Figure 4: Results on House and Hotel sequences using the **third-order model** described in Section 4.1. Other results can be found in the main paper.

Figure 6.

Third-order model. The running time results of the experiments presented in the paper are given in Figure 6.

References

[1] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley. Notes on decomposition methods. *Notes for EE364B, Stanford University*, pages 1–36, 2007. 4

[2] M. Cho, J. Lee, and K. M. Lee. Reweighted random walks for graph matching. In *Computer Vision–ECCV 2010*, pages 492–505. Springer, 2010. 5

[3] M. Cho, J. Sun, O. Duchenne, and J. Ponce. Finding matches in a haystack: A max-pooling strategy for graph matching in the presence of outliers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2083–2090, 2014. 5

[4] L. Condat. Fast projection onto the simplex and the ℓ_1 ball. *Mathematical Programming*, pages 1–11, 2014. 4

[5] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. *Advances in Neural Information Processing Systems*, 19:313, 2007. 5

[6] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2383–2395, 2011. 5

[7] D. K. Lê-Huu and N. Paragios. Alternating direction graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6253–6261, 2017. 1

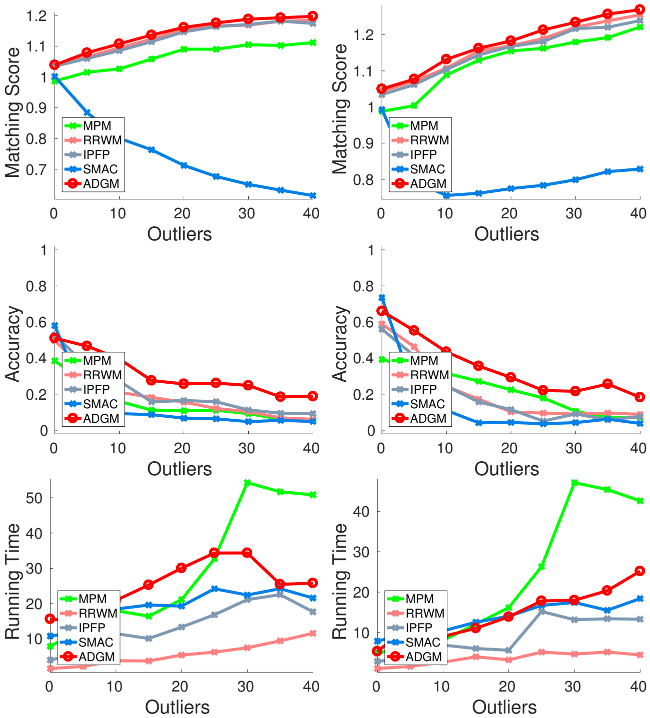
[8] J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted random walks. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1633–1640. IEEE, 2011. 5

[9] M. Leordeanu, M. Hebert, and R. Sukthankar. An integer projected fixed point method for graph matching and map inference. In *Advances in neural information processing systems*, pages 1114–1122, 2009. 5

[10] Q. Nguyen, A. Gautier, and M. Hein. A flexible tensor block coordinate ascent scheme for hypergraph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5270–5278, 2015. 5

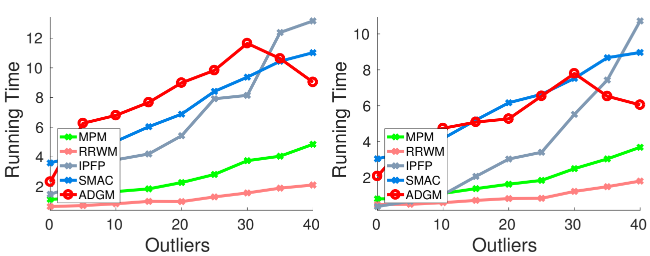
[11] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2002. 4

[12] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 5



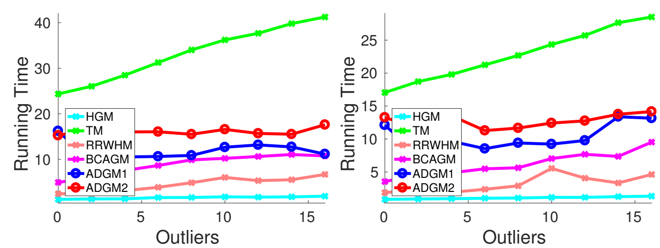
(a) Cars (b) Motorbikes

Figure 5: Results on Cars and Motorbikes using the **pairwise model B** described in Section 4.1. ADGM always achieved the best objective values that are higher than the ground-truth ones. However, the obtained accuracies are still very low. One can conclude that this pairwise model is not suited for this dataset.



(a) Cars (b) Motorbikes

Figure 6: Running time on Cars and Motorbikes using the **pairwise model C** described in Section 4.1.



(a) Cars (b) Motorbikes

Figure 7: Running time on Cars and Motorbikes using the **third-order model** described in Section 4.1.