

# DS++: A Flexible, Scalable and Provably Tight Relaxation for Matching Problems

ANONYMOUS AUTHOR(S)

Correspondence problems are often modelled as quadratic optimization problems over permutations. Common scalable methods for approximating solutions of these NP-hard problems are the spectral relaxation for non-convex energies and the doubly stochastic (DS) relaxation for convex energies. Lately, it has been demonstrated that semidefinite programming relaxations can have considerably improved accuracy at the price of a much higher computational cost.

We present a convex quadratic programming relaxation which is provably stronger than both DS and spectral relaxations, with the same scalability as the DS relaxation. The derivation of the relaxation also naturally suggests a projection method for achieving meaningful integer solutions which improves upon the standard closest-permutation projection. Our method can be easily extended to optimization over doubly stochastic matrices, partial or injective matching, and problems with additional linear constraints. We employ recent advances in optimization of linear-assignment type problems to achieve an efficient algorithm for solving the convex relaxation.

We present experiments indicating that our method is more accurate than local minimization or competing relaxations for non-convex problems. We successfully apply our algorithm to shape matching and to the problem of ordering images in a grid, obtaining results which compare favorably with state-of-the-art methods.

We believe our results indicate that our method should be considered the method of choice for quadratic optimization over permutations.

## ACM Reference format:

Anonymous Author(s). 2017. DS++: A Flexible, Scalable and Provably Tight Relaxation for Matching Problems . 1, 1, Article 1 (May 2017), 14 pages.  
DOI: 10.1145/nnnnnnnn.nnnnnnnn

## 1 INTRODUCTION

Matching problems, seeking some useful correspondence between two shapes or, more generally, discrete metric spaces, are central in computer graphics and vision. Matching problems are often modeled as optimization of a quadratic energy over permutations. Global optimization and approximation of such problems is known to be NP-hard [Loiola et al. 2007].

A common strategy for dealing with the computational hardness of matching problems is replacing the original optimization problem with an easier, similar problem that can be solved globally and efficiently. Perhaps the two most common *scalable* relaxations for such problems are the spectral relaxation for energies with non-negative entries [Feng et al. 2013; Leordeanu and Hebert 2005], and

<sup>0</sup>In accordance with SIGGRAPH policy we state that this submission is available on arxiv [Anonymous 2017]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2017 ACM. XXXX-XXXX/2017/5-ART1 \$15.00

DOI: 10.1145/nnnnnnnn.nnnnnnnn



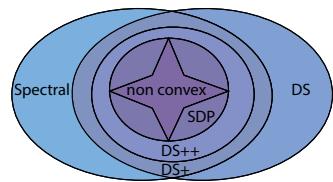
Fig. 1. Our algorithm offers a flexible and scalable framework for matching metric spaces and is guaranteed to perform better than the classical spectral and doubly-stochastic relaxations. Left: non-rigid matching computed automatically between two raw scans with topological issues from the FAUST dataset [Bogo et al. 2014]; Right, an automatic arrangement of natural images in a 2D grid based on deep features-based pairwise affinity. Note how similar objects are clustered together.

the doubly stochastic (DS) relaxation for convex energies [Aflalo et al. 2015; Fiori and Sapiro 2015].

Our work is motivated by the recent work of [Kezurer et al. 2015] and related works in optimization literature [Zhao et al. 1998] who proposed a semi-definite programming (SDP) relaxation which is provably stronger than both spectral and DS relaxations. The obtained relaxation was shown empirically to be extremely tight, achieving the global ground truth in most experiments presented. However, a major limitation was the computational cost of solving a semi-definite program with  $O(n^4)$  variables. Accordingly in this paper we pursue the following question:

**Question:** Is it possible to construct a relaxation which is stronger than the spectral and DS relaxations, without compromising efficiency?

We give an affirmative answer to this question and show that by correctly combining the spectral and DS relaxations in the spirit of [Fogel et al. 2013] we obtain a relaxation which is provably tighter than both, and is in fact in a suitable sense exactly the intersection of both relaxations. We name this relaxation DS+. Moreover, we observe that a refined spectral analysis leads to a significant improvement to this relaxation and a provably tighter quadratic program we name DS++. This relaxation enjoys the same scalability as DS and DS+ as all three are quadratic programs with  $n^2$  variables and the same number of constraints.



Additional time efficiency is provided by specialized solvers for the DS relaxation such as [Solomon et al. 2016]. We note that DS++ is still less tight than the final expensive and accurate relaxation of [Kezurer et al. 2015] yet strikes a balance between tightness and computational complexity. The hierarchy between the relaxations is illustrated in the inset and proven in section 4.

Since DS++ is a relaxation, it is not guaranteed to output an integer solution (*i.e.*, a permutation). To obtain a feasible permutation we propose a homotopy-type method, in the spirit of [Ogier and Beyer 1990; Zaslavskiy et al. 2009]. This method, which continuously deforms the energy functional from convex to concave, is guaranteed to produce an integer-solution and in practice outperforms standard Euclidean projection techniques. Essentially it provides a strategy for finding a local minima for the original non-convex problem using a good initial guess obtained from the convex relaxation.

Our algorithm is very flexible and can be applied to both convex and non-convex energies (in contrast with DS), and to energies combining quadratic and linear terms (in contrast with the spectral relaxation, which also requires energies with non-negative entries). It can also be easily modified to allow for additional linear constraints, injective and partial matching, and solving quadratic optimization problems over the doubly stochastic matrices. We present experiments demonstrating the effectiveness of our method in comparison to random initializations of the non-convex problem, spectral, DS, and DS+ relaxations, as well as lifted linear-programming relaxations.

We have tested our algorithm on three applications: (i) non-rigid matching; (ii) image arrangements; and (iii) coarse-to-fine matching. Comparison to state-of-the-art algorithms for these applications shows that our algorithm produces favorable results in comparable speed.

Our contributions in this paper are threefold:

- (1) We identify the optimal initial convex and concave relaxation.
- (2) We show, both theoretically and experimentally that the proposed algorithm is more accurate than other popular contemporary methods. We believe that establishing a hierarchy between the various relaxation methods for quadratic matching is crucial both for applications, and for pushing forward the algorithmic state of the art, developing stronger optimization algorithms in the future.
- (3) Lastly, we build a simple end-to-end algorithm utilizing recent advances in optimization over the doubly-stochastic matrices to provide a scalable yet accurate algorithm for quadratic matching.

## 2 PREVIOUS WORK

Many works in computer vision and graphics model correspondence problems as quadratic optimization problems over permutation matrices. In many cases these problems emerge as discretizations of isometry-invariant distances between shapes [Mémoli 2011; Mémoli and Sapiro 2005]. We focus here on the different methods to approximately solve these computationally hard problems.

*Spectral relaxation.* The spectral relaxation for correspondence problems in computer vision has been introduced in [Leordeanu and Hebert 2005] and has since become very popular in both computer vision and computer graphics, *e.g.*, [De Aguiar et al. 2008; Feng et al. 2013; Liu et al. 2009; Shao et al. 2013]. This method replaces the requirement for permutation matrices with a single constraint on the Frobenius norm of the matrices to obtain a maximal eigenvalue problem. It requires energies with positive entries to ensure the obtained solution is positive. This relaxation is scalable but is not a very tight approximation of the original problem. A related relaxation appears in [Rodola et al. 2012], where the variable  $x$  is constrained to be non-negative with  $\|x\|_1 = 1$ . This optimization problem is generally non-convex, but the authors suggest a method for locally minimizing this energy to obtain a sparse correspondence.

*DS relaxation.* An alternative approach relaxes the set of permutations to its convex hull of doubly stochastic matrices [Schellewald et al. 2001]. When the quadratic objective is convex, this results in a convex optimization problem (quadratic program) which can be minimized globally, although the minimum may differ from the global minima of the original problem. [Solomon et al. 2012] argue for the usefulness of the fuzzy maps obtained from the relaxation. For example, for symmetric shapes fuzzy maps can encode all symmetries of the shape.

[Aflalo et al. 2015] shows that for the convex graph matching energy the DS relaxation is equivalent to the original problem for generic asymmetric and isomorphic graphs. These results are strengthened in [Fiori and Sapiro 2015]. However when noise is present the relaxations of the convex graph matching energy will generally not be equivalent to the original problem [Lyzinski et al. 2016]. Additionally, for concave energies the DS relaxation is always equivalent to the original problem [Gee and Prager 1994], since minima of concave energies are obtained at extreme points. The challenge for non-convex energies is that global optimization over DS matrices is not tractable.

To achieve good initialization for local minimization of such problems, [Gee and Prager 1994; Ogier and Beyer 1990; Zaslavskiy et al. 2009] suggest to minimize a sequence of energies  $E_t$  which gradually vary from a convex energy  $E_0$  to an equivalent concave energy  $E_1$ . In this paper we adopt this strategy to obtain an integer solution, and improve upon it by identifying the optimal convex and concave energies from within the energies  $E_t$ .

The authors of [Fogel et al. 2013, 2015] show that the DS relaxation can be made more accurate by adding a concave penalty of the form  $-a \|X\|_F^2$  to the objective. To ensure the objective remains convex they suggest to choose  $a$  to be the minimal eigenvalue of the quadratic objective. We improve upon this choice by choosing  $a$  to be the minimal eigenvalue *over the doubly stochastic subspace*, leading to a provably tighter relaxation. The practical advantage of our choice (DS++) versus Fogel's choice (DS+) is significant in terms of the relaxation accuracy as demonstrated later on. The observation that this choice suffices to ensure convexity has been made in the convergence proof of the softassign algorithm [Rangarajan et al. 1997].

*Optimization of DS relaxation.* Specialized methods for minimization of linear energies over DS matrices [Benamou et al. 2015; Cuturi 2013; Kosowsky and Yuille 1994; Solomon et al. 2015] using entropic regularization and the Sinkhorn algorithm are considerably more efficient than standard linear program solvers for this class of problems. Motivated by this, [Rangarajan et al. 1996] propose an algorithm for globally minimizing quadratic energies over doubly stochastic matrices by iteratively minimizing regularized linear energies using Sinkhorn type algorithms. For the optimization in this paper we applied [Solomon et al. 2016] who offer a different algorithm for locally minimizing the Gromov-Wasserstein distance by iteratively solving regularized linear programs. The advantage of the latter algorithm over the former algorithm is its certified convergence to a critical point when applied to non-convex quadratic energies.

*Other convex relaxations.* Stronger relaxations than the DS relaxation can be obtained by lifting methods which add auxiliary variables representing quadratic monomials in the original variables. This enables adding additional convex constraints on the lifted variables. A disadvantage of these methods is the large number of variables which leads to poor scalability. [Kezurer et al. 2015] propose an SDP relaxation in the spirit of [Zhao et al. 1998], and prove it is stronger than both DS (for convex objective) and spectral relaxations, and in practice often achieves the global minimum of the original problem. However, it is only tractable for up to fifteen points. [Chen and Koltun 2015] use a lifted linear program relaxation in the spirit of [Adams and Johnson 1994; Werner 2007]. To deal with scalability issues they use Markov random field techniques [Kolmogorov 2006] to approximate the solution of their linear programming relaxation.

*Quadratic assignment.* Several works aim at globally solving the quadratic assignment problem using combinatorial methods such as branch and bound. According to a recent survey [Loiola et al. 2007] these methods are not tractable for graphs with more than 30 points. Branch and bound methods are also in need of convex relaxation to achieve lower bounds for the optimization problem. [Anstreicher and Brixius 2001] provide a quadratic programming relaxation for the quadratic assignment problem which provably achieves better lower bounds than a competing spectral relaxation using a method which combines spectral, linear, and DS relaxations. Improved lower bounds can be obtained using second order cone programming [Xia 2008] and semi-definite programming [Ding and Wolkowicz 2009] in  $O(n^2)$  variables. All the relaxations above use the specific structure of the quadratic assignment problem while our relaxation is applicable to general quadratic objectives which do not carry this structure and are very common in computer graphics. For example, most of the correspondence energies formulated below and considered in this paper cannot be formulated using the quadratic assignment energy.

*Other approaches for shape matching.* A similar approach to the quadratic optimization approach is the functional map method (e.g., [Ovsjanikov et al. 2012]) which solves a quadratic optimization problem over permutations and rotation matrices, typically using high-dimensional ICP provided with some reasonable initialization.

Recently [Maron et al. 2016] proposed an SDP relaxation for this problem with considerably improved scalability with respect to standard SDP relaxations.

Supervised learning techniques have been successfully applied for matching specific classes of shapes in [Masci et al. 2015; Rodolà et al. 2014; Wei et al. 2016; Zuffi and Black 2015]. A different approach for matching near isometric shapes is searching for a mapping in the low dimensional space of conformal maps which contains the space of isometric maps [Kim et al. 2011; Lipman and Funkhouser 2009; Zeng et al. 2010]. More information on shape matching can be found in shape matching surveys such as [Van Kaick et al. 2011].

### 3 APPROACH

*Motivation.* Quadratic optimization problems over the set of permutation matrices arise in many contexts. Our main motivating example is the problem of finding correspondences between two metric spaces (e.g., shapes)  $(\mathcal{S}, d_{\mathcal{S}})$  and  $(\mathcal{T}, d_{\mathcal{T}})$  which are related by a perfect or an approximate isometry. This problem can be modeled by uniformly sampling the spaces to obtain  $\{\mathbf{s}_1, \dots, \mathbf{s}_n\} \subseteq \mathcal{S}$  and  $\{\mathbf{t}_1, \dots, \mathbf{t}_n\} \subseteq \mathcal{T}$ , and then finding the permutation  $X \in \Pi_n$  which minimizes an energy of the form

$$E(X) = \sum_{ijk\ell} W_{ijk\ell} X_{ij} X_{k\ell} + \sum_{ij} C_{ij} X_{ij}. \quad (1)$$

Here  $W_{ijk\ell}$  is some penalty on deviation from isometry: If the points  $\mathbf{s}_i, \mathbf{s}_k$  correspond to the points  $\mathbf{t}_j, \mathbf{t}_\ell$  (resp.), then the distances between the pair on the source shape and the pair on the target shape should be similar. Therefore we choose

$$W_{ijk\ell} = p(d_{\mathcal{S}}(\mathbf{s}_i, \mathbf{s}_k), d_{\mathcal{T}}(\mathbf{t}_j, \mathbf{t}_\ell)) \quad (2)$$

where  $p(u, v)$  is some function penalizing for deviation from the set  $\{(u, v) \mid u = v\} \subseteq \mathbb{R}^2$ . Several different choices of  $p$  exist in the literature.

The linear term  $C$  is sometimes used to aid the correspondence task by encouraging correspondences  $\mathbf{s}_i \mapsto \mathbf{t}_j$  between points with similar isometric-invariant descriptors.

*Problem statement.* Our goal is to solve quadratic optimization problems over the set of permutations as formulated in (1). Denoting the column stack of permutations  $X \in \mathbb{R}^{n \times n}$  by the vector

$$x = [X_{11}, X_{21}, \dots, X_{nn}]^T \in \mathbb{R}^{n^2}$$

leads to a more convenient phrasing of (1):

$$\min_X E(X) = x^T W x + c^T x + d \quad (3a)$$

$$\text{s.t. } X \in \Pi_n \quad (3b)$$

This optimization problem is non-convex for two reasons. The first is the non-convexity of  $\Pi_n$  (as a discrete set of matrices), and the second is that  $E$  is often non-convex (if  $W$  is not positive-definite). As global minimization of (3) is NP-hard [Loiola et al. 2007] we will be satisfied with obtaining a good approximation to the global solution of (3) using a scalable optimization algorithm. We do this by means of a convex relaxation coupled with a suitable projection algorithm for achieving integer solutions.

### 1 3.1 Convex relaxation

2 We formulate our convex relaxation by first considering a one-  
 3 parameter family of equivalent formulations to (3): observe that for  
 4 any permutation matrix  $X$  we have that  $\|X\|_F^2 = n$ . It follows that  
 5 all energies of the form

$$6 E(X, a) = E(X) - a \|X\|_F^2 + a \cdot n \quad (4)$$

7 coincide when restricted to the set of permutations. Therefore, replacing  
 8 the energy in (3) with  $E(X, a)$  provides a one-parameter family  
 9 of equivalent formulations. For some choices of  $a$  the energy in  
 10 these formulations is convex, for example, for any  $a \leq \lambda_{\min}$ , where  
 11  $\lambda_{\min}$  is the minimal eigenvalue of  $W$ .

12 For each such equivalent formulation we consider its *doubly*  
 13 *stochastic* relaxation. That is, replacing the permutation constraint  
 14 (3b) with its convex-hull, the set of doubly-stochastic matrices:

$$15 \min_X E(X, a) \quad (5a)$$

$$16 \text{ s.t. } X\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^T X = \mathbf{1}^T \quad (5b)$$

$$17 \quad X \geq 0 \quad (5c)$$

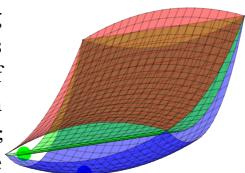
18 Our goal is to pick a relaxed formulation (*i.e.*, choose an  $a$ ) that  
 19 provides the best lower bound to the global minimum of the original  
 20 problem (3). For that end we need to consider values of  $a$  that make  
 21  $E(X, a)$  convex and consequently turn (5) into a convex program  
 22 that provide a lower bound to the global minimum of (3).

23 Among all the convex programs described above we would like  
 24 to choose the one which provides the tightest lower bound. We will  
 25 use the following simple lemma proved in Appendix A:

26 **LEMMA 3.1.** *For all doubly stochastic  $X$  we have  $E(X, a) \leq E(X, b)$   
 27 when  $a < b$ , and  $E(X, a) = E(X, b)$  if and only if  $X$  is a permutation.*

28 An immediate conclusion from this lemma is that  
 29  $\min_{X \in DS} E(X, a) \leq \min_{X \in DS} E(X, b)$  and so the best lower  
 30 bound will be provided by the largest value of  $b$  for which  $E(X, b)$   
 31 is convex.

32 See for example the inset illustrating  
 33 the energy graphs for different  $a$  values  
 34 for a toy-example: in red - the graph of  
 35 the original (non-convex) energy with  
 36  $a = 0$ ; in blue the energy with  $a < \lambda_{\min}$ ;  
 37 and in green  $a = \lambda_{\min}$ . Note that the  
 38 green graph lies above the blue graph  
 39 and all graphs coincide on the corners (*i.e.*, at the permutations).  
 40 Since the higher the energy graph the better lower bound we achieve  
 41 it is desirable to take the maximal  $a$  that still provides a convex  
 42 program in (5). In the inset the green and blue points indicate  
 43 the solution of the respective relaxed problems; in this case the green  
 44 point is much closer to the sought after solution, *i.e.*, the lower-left  
 45 corner.



46 To summarize the above discussion: the desired  $a$  is the maximal  
 47 value for which  $E(X, a)$  is a convex function. As noted above choosing  
 48  $a = \lambda_{\min}$  in the spirit of [Fogel et al. 2013], leads to a convex  
 49 problem which we denote by DS+. However this is in fact not the  
 50 maximal value in general.

52 To find the maximal  $a$  we can utilize the fact that  $X$  is constrained  
 53 to the affine space defined by the constraints (5b): We parameterize  
 54 the affine space as  $x = x_0 + Fz$ , where  $x_0$  is some permutation,  $F$  is  
 55 any parameterization satisfying  $F^T F = I$ , and  $z \in \mathbb{R}^{(n-1)^2}$ . Plugging  
 56 this into  $E(X, a)$  provides a quadratic function in  $z$  of the form

$$57 z^T F^T (W - aI) Fz + \text{aff}(z)$$

58 where  $\text{aff}(z)$  is some affine function of  $z$ . It follows that (5) will be  
 59 convex iff  $F^T (W - aI) F$  is positive semi-definite. The largest possible  
 60  $a$  fulfilling this condition is the minimal eigenvalue of  $F^T WF$  which  
 61 we denote by  $\bar{\lambda}_{\min}$ . Thus our convex relaxation which we name  
 62 DS++ boils down to minimizing (5) with  $a = \bar{\lambda}_{\min}$ .

### 63 3.2 Projection

64 We now describe our method for projecting the solution of our  
 65 relaxation onto the set of permutations. This method is inspired by  
 66 the "convex to concave" method from [Gee and Prager 1994; Ogier  
 67 and Beyer 1990; Zaslavskiy et al. 2009], but also improves upon  
 68 these works by identifying the correct interval on which the convex  
 69 to concave procedure should be applied as we now describe.

70 Lemma 3.1 tells us that the global optimum of  $E(X, a)$  over the  
 71 doubly stochastic matrices provides an increasingly better approxi-  
 72 mation of the global optimum of the original problem (3) as we  
 73 keep increasing  $a$  even beyond the convex regime, that is  $a > \bar{\lambda}_{\min}$ .  
 74 In fact, it turns out that if  $a$  is chosen large enough so that  $E(X, a)$   
 75 is strictly *concave*, then the global optima of (5) and the global optima  
 76 of the original problem over permutations are identical. This  
 77 is because the (local or global) minima of strictly concave functions  
 78 on a compact convex set are always obtained at the extreme points  
 79 of the set. In our case, the permutations are these extreme points.

80 This leads to a natural approach to approximate the global opti-  
 81 mum of (3): Solve the above convex problem with  $a = \bar{\lambda}_{\min}$  and  
 82 then start increasing  $a > \bar{\lambda}_{\min}$  until an integer solution is found.  
 83 We choose a finite sequence  $a_0 < a_1 < \dots < a_N$ , where  $a_0 = \bar{\lambda}_{\min}$   
 84 and  $E(X, a_N)$  is strictly concave. We begin by solving (5) with  $a_0$   
 85 which is exactly the convex relaxation described above and obtain a  
 86 minimizer  $X_0$ . We then iteratively locally minimize (5) with  $a = a_i$   
 87 using as an initialization the previous solution  $X_{i-1}$ . The reasoning  
 88 behind this strategy is that when  $a_i$  and  $a_{i-1}$  are close a good solu-  
 89 tion for the latter should provide a good initialization for the former,  
 90 so that at the end of the process we obtain a good initial guess for  
 91 the minimization of  $E(X, a_N)$ , which is equivalent to the original  
 92 integer program. We stress that although the obtained solution may  
 93 only be a local minimum, it will necessarily be a permutation.

94 To ensure that  $E(X, a_N)$  is strictly concave we can choose any  
 95  $a_N$  larger than  $\bar{\lambda}_{\max}$ , which analogously to  $\bar{\lambda}_{\min}$  is defined as the  
 96 largest eigenvalue of  $F^T WF$ . In practice we select  $a_N = \bar{\lambda}_{\max}$  which  
 97 in the experiments we conducted is sufficient for obtaining integer  
 98 solutions. We then took  $a_i$  by uniformly sampling  $[a_0, a_N]$  where  
 99 unless stated otherwise we used ten samplings ( $N = 9$ ). Throughout  
 100 the paper we will use the term *DS++ algorithm* to refer to our com-  
 101 plete method (relaxation+projection) and DS++ or DS++ relaxation  
 102 to refer only to the relaxation component.

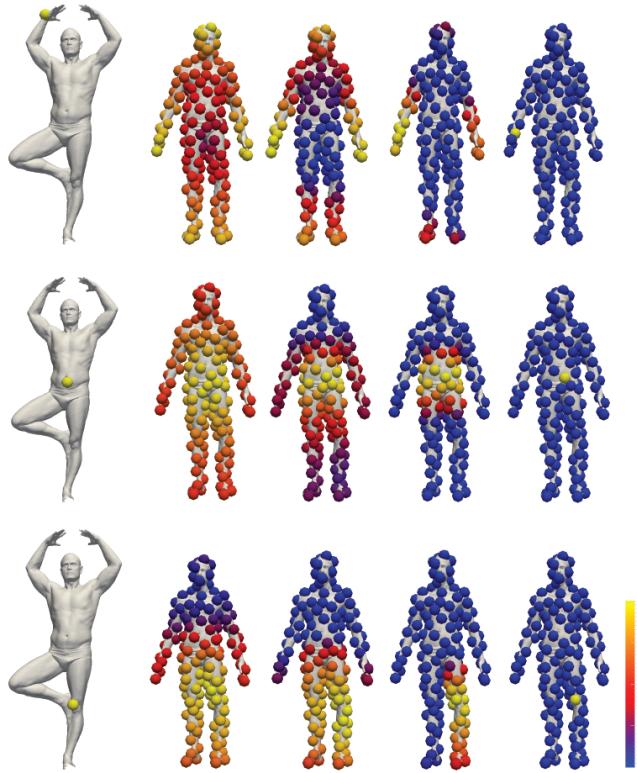


Fig. 2. Visualization of the projection procedure. For each point on the source (left) a fuzzy correspondence is obtained by minimizing the convex energy (second from the left). The correspondence gradually becomes sharper as the projection procedure proceeds until the final step of minimizing a concave energy where a well defined map is obtained (right).

Figure 2 shows the correspondences (encoded in a specific row of  $X$ ) obtained at different stages of the projection procedure when running our algorithm on the FAUST dataset [Bogo et al. 2014] as described below. The figure shows the correspondences obtained from optimizing  $E(X, a_i)$  for  $i = 0, 4, 7, N = 9$ .

Our algorithm is summarized in Algorithm 1: In Section 5 we discuss efficient methods for implementing this algorithm.

---

**Algorithm 1:** DS++ algorithm
 

---

**Input:** The energy components  $W, c, d$

Compute  $\bar{\lambda}_{\min}, \bar{\lambda}_{\max}$  of  $F^T W F$ ;

Choose  $N + 1$  uniform samples  $a_0 = \bar{\lambda}_{\min}, a_1, \dots, a_N = \bar{\lambda}_{\max}$ ;

Solve (5) with  $a = a_0$  to obtain  $X_0$  ;

**for**  $i = 1 \dots N$  **do**

Solve (5) with  $a = a_i$  initialized from  $X_{i-1}$  to obtain  $X_i$  ;

**Output:** The permutation  $X_N$

---

## 4 COMPARISON WITH OTHER RELAXATIONS

The purpose of this section is to theoretically compare our relaxation with common competing relaxations. We prove

**THEOREM 4.1.** *The DS++ relaxation is more accurate than DS+, which in turn is more accurate than the spectral and doubly stochastic relaxation.*

*The SDP relaxation of [Kezurer et al. 2015] is more accurate than all the relaxations mentioned above.*

Our strategy for proving this claim is formulating all relaxations in a unified framework, using the SDP lifting technique in [Kezurer et al. 2015], that in turn readily enables comparison of the different relaxations.

The first step in constructing SDP relaxations is transforming the original problem (3) into an equivalent optimization problem in a higher dimension. The higher dimension problem is formulated over the set:

$$\Pi_n^\uparrow = \left\{ (X, Y) \mid X \in \Pi_n, \quad Y = xx^T \right\}$$

Using the identity

$$\text{tr} W Y = \text{tr} W x x^T = x^T W x$$

we obtain an equivalent formulation to (3):

$$\begin{aligned} \min_{X, Y} \quad & E(X, Y) = \text{tr} W Y + c^T x + d \\ \text{s.t.} \quad & (X, Y) \in \Pi_n^\uparrow \end{aligned}$$

SDP relaxations are constructed by relaxing the constraint  $(X, Y) \in \Pi_n^\uparrow$  using linear constraints on  $X, Y$  and the semi-definite constraint  $Y \succeq xx^T$ .

[Kezurer et al. 2015] showed that the spectral and doubly stochastic relaxations are equivalent to the following SDP relaxations:

$$\begin{array}{ll} \max & E(X, Y) \\ (\mathcal{S}^\dagger) \quad \text{s.t.} & \text{tr } Y = n \\ & Y \succeq xx^T \end{array} \quad \begin{array}{ll} \max & E(X, Y) \\ (\mathcal{DS}^\dagger) \quad \text{s.t.} & X \in \mathcal{DS} \\ & Y \succeq xx^T \end{array}$$

We note that the spectral relaxation is applicable only when  $c = 0$ , and the DS relaxation is tractable only when the objective is convex, i.e.,  $W \succeq 0$ . The equivalence holds under these assumptions.

Given this new formulation of spectral and DS, an immediate method for improving both relaxations is considering the *Intersection-SDP*, obtained by enforcing the constraints from both  $(\mathcal{DS}^\dagger)$  and  $(\mathcal{S}^\dagger)$ . The relaxation can be further improved by adding additional linear constraints on  $(X, Y)$ . This is the strategy followed by [Kezurer et al. 2015] to achieve their final tight relaxation which is presented in Eq. (9) in the appendix. The main limitation of this approach is its prohibitive computational price resulting from solving SDPs with  $O(n^4)$  variables, in strong contrast to the original formulation of spectral and DS that uses only  $n^2$  variables (i.e., the permutation  $X$ ). This naturally leads to the research question we posed in the introduction, which we can now state in more detail:

**Question:** Is it possible to construct an SDP relaxation which is stronger than  $(\mathcal{DS}^\dagger)$  and  $(\mathcal{S}^\dagger)$ , and yet is equivalent to a tractable and scalable optimization problem with  $n^2$  variables?



Fig. 3. Typical maps obtained using our method on the FAUST dataset [Bogo et al. 2014]. In each pair: left mesh is colored linearly and the computed map is used to transfer the coloring to the target, right mesh.



Fig. 4. Image arrangement according to the mean color of images using the DS++ algorithm. Table 1 shows corresponding quantitative results.

We answer this question affirmatively by showing that the Intersection-SDP is in fact equivalent to DS+. Additionally DS++ is equivalent to a stronger SDP relaxation which includes all constraints from the *Intersection-SDP*, as well as the following additional  $2n^3$  constraints: Let us write the linear equality constraints appearing in the definition of the DS matrices (*i.e.*, (5b)) in the form  $Ax = b$ . Then any  $(X, Y) \in \Pi_n^\uparrow$  in particular satisfies  $Axx^T = bx^T$  and therefore also:  $AY = bx^T$ . Adding these constraints to the Intersection-SDP we obtain

$$\min_{X, Y} E(X, Y) \quad (6a)$$

$$\text{s.t. } \text{tr}Y = n \quad (6b)$$

$$X \geq 0 \quad (6c)$$

$$Ax = b \quad (6d)$$

$$AY = bx^T \quad (6e)$$

$$Y \geq xx^T \quad (6f)$$

Theorem 4.1 now follows from:

- LEMMA 4.2. (1) *The Intersection-SDP is equivalent to DS+.*  
 (2) *The SDP relaxation in (6) is equivalent to DS++.*  
 (3) *The SDP relaxation of [Kezurer et al. 2015] can be obtained by adding additional linear constraints to (6).*

We prove the lemma in the appendix.

## 5 IMPLEMENTATION DETAILS

*Entropic regularization.* Optimization of (5) can be done using general purpose non-convex solvers such as Matlab's *fmincon*, or solvers for convex and non-convex quadratic programs. We opted for the recent method of Solomon et al., [2016] that introduced a specialized scalable solver for local minimization of regularized quadratic functionals over the set of doubly stochastic matrices.

The algorithm of [Solomon et al. 2016] is based on an efficient algorithm for optimizing the KL divergence

$$KL(x|y) = \langle x, \log x \rangle - \langle x, \log y \rangle$$

where  $x$  is the column stack of a doubly stochastic matrix  $X$  and  $y$  is some fixed positive vector. The solution for the KKT equations of this problem can be obtained analytically for  $x$ , up to scaling of the rows and columns, which is performed by the efficient Sinkhorn algorithm. See [Cuturi 2013] for more details.

The algorithm of [Solomon et al. 2016] minimizes quadratic functionals  $f(x) = x^T H x + c^T x$  (where in our case  $H = W - aI$ ) over doubly stochastic matrices by iteratively optimizing KL-divergence problems. First the original quadratic functional is regularized by adding a barrier function  $\alpha \langle x, \log x \rangle$  keeping the entries of  $x$  away from zero to obtain a new functional

$$f_\alpha(x) = f(x) + \alpha \langle x, \log x \rangle$$

The parameter  $\alpha$  is chosen to be some small positive number so that its effect on the functional is small. We then define  $g_\alpha(x) =$

1  $\exp(-\alpha^{-1}(Hx + c))$  so that  
 2  
 3

$$f_\alpha(x) = \alpha KL(x|g_\alpha(x))$$

4 We then optimize  $f_\alpha$  iteratively: In iteration  $k+1$ ,  $g_\alpha$  is held fixed at  
 5 its previous value  $x = x_k$ , and an additional term  $KL(x|x_k)$  is added  
 6 penalizing large deviations of  $x$  from  $x_k$ . More precisely,  $x_{k+1}$  is  
 7 defined to be the minimizer of

$$8 \eta KL(x|g_\alpha(x_k)) + (1 - \eta)KL(x|x_k) = KL(x|g_\alpha^\eta(x_k) \odot x_k^{1-\eta})$$

9 where  $\odot$  denotes entry-wise multiplication of vectors. For small  
 10 enough values of  $\eta$ , [Solomon et al. 2016] prove that the algorithm  
 11 converges to a local minimum of  $f_\alpha(x)$ .

12 In our implementation we use  $\eta = 0.01$ . We choose the smallest  
 13 possible  $\alpha$  so that all entries of the argument of the exponent in the  
 14 definition of  $g_\alpha$  are in  $[-100, 100]$ . This choice is motivated by the  
 15 requirement of choosing small  $\alpha$  coupled with the breakdown of  
 16 matlab’s exponent function at around  $e^{700}$ . Note that this choice  
 17 requires  $\alpha = \alpha_k$  to update at each iteration. We find that with this  
 18 choice of  $\alpha$  the regularization term has little effect on the energy  
 19 and we obtain final solutions which are close to being permutations.  
 20 To achieve a perfect permutation we project the final solution using  
 21 the  $L_2$  projection. The  $L_2$  projection is computed by minimizing a  
 22 linear program as described, e.g., in [Zaslavskiy et al. 2009].

23 Computing  $\bar{\lambda}_{\min}$  and  $\bar{\lambda}_{\max}$ . We compute  $\bar{\lambda}_{\min}$  and  $\bar{\lambda}_{\max}$  by solving  
 24 two maximal magnitude eigenvalue problems: We first solve for the  
 25 maximal magnitude eigenvalue of  $F^T WF$ . If this eigenvalue is positive  
 26 then it is equal to  $\bar{\lambda}_{\max}$ . We can then find  $\bar{\lambda}_{\min}$  by translating our  
 27 matrix by  $\bar{\lambda}_{\max}$  to obtain a positive-definite matrix  $\bar{\lambda}_{\max}I - F^T WF$   
 28 whose maximal eigenvalue  $\eta$  is related to the minimal eigenvalue  
 29 of the original matrix via  $\bar{\lambda}_{\min} = \bar{\lambda}_{\max} - \eta$ .

30 If the solution of the first maximal magnitude problem is negative  
 31 then this eigenvalue is  $\bar{\lambda}_{\min}$ , and we can use a process similar to the  
 32 one described above to obtain  $\bar{\lambda}_{\max}$ .

33 Solving maximal magnitude eigenvalue problems requires re-  
 34 peated multiplication of vectors  $v \in \mathbb{R}^{(n-1)^2}$  by the matrix  $F^T WF$ ,  
 35 where  $W \in \mathbb{R}^{n^2 \times n^2}$  and  $F \in \mathbb{R}^{n^2 \times (n-1)^2}$ . If  $W$  is sparse, computing  
 36  $Fv$  can become a computational bottleneck. To avoid this problem,  
 37 we note that  $F^T WF$  has the same maximal eigenvalue as the matrix  
 38  $FF^T WFF^T$  and so compute the maximal eigenvalue of the latter  
 39 matrix. The advantage of this is that multiplication by the matrix  
 40  $P = FF^T$  can be computed efficiently:

41 Since  $P$  is the orthogonal projection onto  $\text{Image}(F)$ , we can use  
 42 the identity  $Pu = u - P_\perp u$  where  $P_\perp$  is the projection onto the  
 43 orthogonal complement of  $\text{Image}(F)$ . The orthogonal complement  
 44 is of dimension  $2n - 1$  and therefore  $P_\perp u = F_\perp F_\perp^T u$  where  $F_\perp \in$   
 45  $\mathbb{R}^{n^2 \times (2n-1)}$ .

46 We solve the maximal magnitude eigenvalue problems using  
 47 Matlab’s function *eigs*.

## 51 6 GENERALIZATIONS

52 *Injective matching.* Our method can be applied with minor  
 53 changes to injective matching. The input of injective matching is  $k$   
 54 points sampled from the source shape  $\mathcal{S}$  and  $n > k$  points sampled  
 55 from the target shape  $\mathcal{T}$ , and the goal is to match the  $k$  points from  
 56  $\mathcal{S}$  injectively to a subset of  $\mathcal{T}$  of size  $k$ .

57 Matrices  $X \in \mathbb{R}^{k \times n}$  representing injective matching have entries  
 58 in  $\{0, 1\}$ , and have a unique unit entry in each row, and at most  
 59 one unit entry in each column. This set can be relaxed using the  
 60 constraints:

$$X\mathbf{1} = \mathbf{1} \quad , \quad \mathbf{1}^T X \leq \mathbf{1}^T \quad (7a)$$

$$\mathbf{1}^T X \mathbf{1} = k \quad (7b)$$

$$X \geq 0 \quad (7c)$$

61 We now add a row with positive entries to the variable matrix  $X$   
 62 to obtain a matrix  $\bar{X} \in \mathbb{R}^{(k+1) \times n}$ . The original matrix  $X$  satisfies  
 63 the injective constraints described above if  $\bar{X}$  satisfies

$$64 \bar{X}\mathbf{1} = (n_1 - k, 1, \dots, 1)^T, \quad \mathbf{1}^T \bar{X} = \mathbf{1}^T \\ 65 \bar{X} \geq 0$$

66 These constraints are identical to the constraint defining DS, up to  
 67 the value of the marginals which have no affect on our algorithm.  
 68 As a result we can solve injective matching problems without any  
 69 modification of our framework.

70 *Partial matching.* The input of partial matching is  $n_1, n_2$  points  
 71 sampled from the source and target shape, and the goal is to match  
 72  $k \leq n_1, n_2$  points from  $\mathcal{S}$  injectively to a subset of  $\mathcal{T}$  of size  $k$ . We  
 73 do not pursue this problem in this paper as we did not find partial  
 74 matching necessary for our applications. However we believe our  
 75 framework can be applied to such problems by adding a row and  
 76 column to the matching matrix  $X$ .

77 *Adding linear constraints.* Modeling of different matching prob-  
 78 lems can suggest adding additional linear constraints on  $X$  that  
 79 can be added directly to our optimization technique. Additional  
 80 linear equality constraints further decrease the dimension of the  
 81 affine space  $X$  is constrained to and as a result make the interval  
 82  $[\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$  smaller, leading to more accurate optimization. We note  
 83 however that incorporating linear constraints into the optimization  
 84 method of [Solomon et al. 2016] is not straightforward.

85 *Upsampling.* Upsampling refers to the task of interpolating corre-  
 86 spondences between  $r$  points sampled from source and target metric  
 87 spaces to a match between a finer sampling of  $k >> r$  source points  
 88 and  $n \geq k$  target points. We suggest two strategies for this problem:  
 89 *Limited support interpolation* and *greedy interpolation*.

90 Limited support interpolation uses the initially matched  $r$  points  
 91 to rule out correspondences between the finely sampled points.  
 92 The method of ruling out correspondences is discussed in the  
 93 Appendix. We enforce the obtained sparsity pattern by writing  
 94  $X = X_{\text{permissible}} + X_{\text{forbidden}}$ , where the first matrix is zero in all  
 95 forbidden entries and the second is zero in all permissible entries.  
 96 We then minimize the original energy  $E(X)$  only on the permissible  
 97 entries, and add a quadratic penalty for the forbidden entries. That  
 98 is, we minimize

$$99 E(X_{\text{permissible}}) + \rho \|X_{\text{forbidden}}\|_F^2$$

100 choosing some large  $\rho > 0$ . The sparsity of  $X_{\text{permissible}}$  enables  
 101 minimizing this energy for large  $k, n$  for which minimizing the  
 102 original energy is intractable.

When  $k, n$  are large we use greedy interpolation. We match each source point  $s_i$  separately. We do this by optimizing over correspondences between  $r + 1$  source points and  $n$  target points, where the  $r + 1$  points are the  $r$  known points and the point  $s_i$ . Since there are only  $n - r$  such correspondences optimization can be performed globally by checking all possible correspondences.

*Optimization over doubly stochastic matrices.* Our main focus was on optimization problems over permutations. However in certain cases the requested output from the optimization algorithm may be a doubly stochastic matrix and not a permutation. When the energy  $E$  is non convex this still remains a non-convex problem. For such optimization problems our method can be applied by taking samples  $a_i$  from the interval  $[\bar{\lambda}_{\min}, 0]$ , since minimization of (5) with  $a = 0$  is the problem to be solved while minimization of (5) with  $a = \bar{\lambda}_{\max}$  forces a permutation solution.

## 7 EVALUATION

In this section we evaluate our algorithm and compare its performance with relevant state of the art algorithms. We ran all experiments on the 100 mesh pairs from the FAUST dataset [Bogo et al. 2014] which were used in the evaluation protocol of [Chen and Koltun 2015].

*Comparison with [Solomon et al. 2016].* In figure 5(b) we compare our method for minimizing non-convex functionals with the local minimization algorithm of [Solomon et al. 2016]. Since this method is aimed at solving non-convex functionals over doubly-stochastic matrices, we run our algorithm using samples in  $[\bar{\lambda}_{\min}, 0]$  as explained in Section 6. We sample 200 points from each mesh using farthest point sampling [Eldar et al. 1997], and optimize the Gromov-Wasserstein (GW) functional advocated in [Solomon et al. 2016], which amounts to choosing  $p$  from (2) to be  $p(u, v) = (u - v)^2$ . As local minimization depends on initialization we locally minimize 1000 times per mesh pair, using 1000 different random initializations. The initializations are obtained by randomly generating a positive matrix in  $\mathbb{R}^{200 \times 200}$  with uniform distribution, and projecting the result onto the doubly stochastic matrices using the Sinkhorn algorithm. As can be seen in the figure our algorithm, using only ten iterations, was more accurate than all the local minima found using random initializations. As a baseline for comparison we note that the difference in energy between randomly drawn permutations and our solution was around 5000, while the difference in energy shown in the graph is around 500.

*Projection evaluation.* In figure 5(a) we examine how the result obtained from our projection method is influenced by the number of points  $N$  sampled from  $[\bar{\lambda}_{\min}, \bar{\lambda}_{\max}]$ . We compared the behavior of our relaxation with several different choices of  $N$  as well as with the standard  $L_2$  projection onto the set of permutations. As expected, our projection is always better than the  $L_2$  projection, and the projection improves as the number of samples is increased.

*Comparison with other relaxations.* We compare our method with other relaxation based techniques. In figure 5 (c)-(d) we compare our relaxation with the spectral relaxation, the DS+ relaxation, and the SDP relaxation of [Kezurer et al. 2015]. In this experiment the energy we use is non-convex so DS is not applicable.

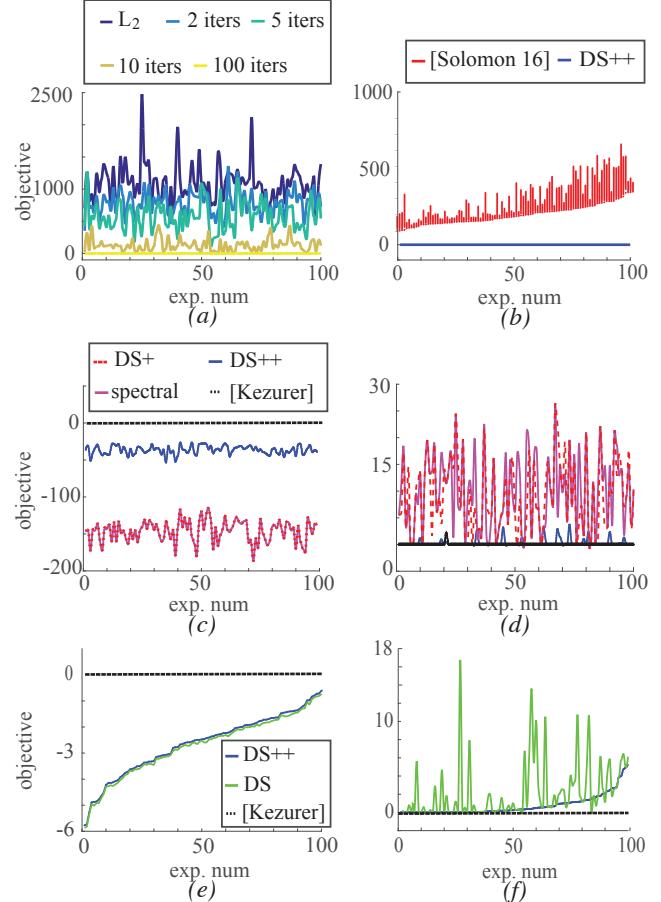


Fig. 5. Evaluation of our algorithm. (a) compares the  $L_2$  projection with our projection. Even with only two iterations our projection improves upon the  $L_2$  projection. Additional iterations yield better accuracy at the price of time complexity. (b) compares minimization of the Gromov-Wasserstein distance with our algorithm and [Solomon et al. 2016] with 1000 random initializations. In all cases we attain a lower objective value. The second row compares lower bounds (c) and upper bounds (d) obtained by the DS++ algorithm, DS+, spectral, and [Kezurer et al. 2015]. As predicted by Theorem 4.1 our lower bounds and upper bounds are outperformed by [Kezurer et al. 2015] who are able to attain the ground truth in these cases, but improve upon those of the remaining methods. The third row compares lower bounds (e) and upper bounds (f) obtained by the DS++ algorithm, DS and [Kezurer et al. 2015] for the convex graph matching functional. The lower bound of the DS++ algorithm modestly improves DS's, while the upper bounds substantially improves the upper bounds of DS's  $L_2$  projection.

We sampled 10 points from both meshes, and minimized the (non-convex) functional selected by [Kezurer et al. 2015], which amounts to choosing  $p$  from (2) to be

$$p(u, v) = -\exp\left(\frac{-(u - v)^2}{\sigma^2}\right)$$

We choose the parameter  $\sigma = 0.2$ . For the minimization we used all four relaxations, obtaining a lower bound for the optimal value, Figure 5 (c). We then projected the solutions obtained onto the set

of permutations, thus obtaining an upper bound, Figure 5 (d). For methods other than the DS++ algorithm we used the  $L_2$  projection. In all experiments the upper and lower bounds provided by the SDP relaxation of [Kezurer et al. 2015] were identical, thus proving that the SDP relaxation found the globally optimal solution. Additionally, in all experiments the upper bound and lower bound provided by our relaxation were superior to those provided by the spectral method, and our projection attained the global minimum in approximately 80% of the experiments in contrast to 11% obtained by the  $L_2$  projection of the spectral method. The differences between the spectral relaxation and the stronger DS+ relaxation were found to be negligible.

In figure 5(e)-(f) we perform the same experiment, but now we minimize the convex graph matching functional  $E(X) = \|AX - XB\|_F^2$  from [Aflalo et al. 2015] for which the classical DS relaxation is applicable. Here again the ground truth is achieved by the SDP relaxation. Our relaxation can be seen to modestly improve the lower bound obtained by the classical DS relaxation, while our projection method substantially improves upon the standard projection.

## 8 APPLICATIONS

We have tested our method for three applications: non-rigid shape matching, image arrangement, and coarse-to-fine matching.

*Non-rigid matching.* We evaluated the performance of our algorithm for non-rigid matching on the high-resolution scans of the FAUST dataset [Bogo et al. 2014]. This is a challenging task since these models are not watertight and contain topological inconsistencies. We compared to [Chen and Koltun 2015] which demonstrated superb state of the art results on this dataset (for non learning-based methods). For a fair comparison we used an identical pipeline to [Chen and Koltun 2015], including their isometric energy modeling and extrinsic regularization term. We first use the DS++ algorithm to match  $n = 160, k = 150$  points, then upsampled to  $n = 450, k = 420$  using limited support interpolation and to  $n = 5000, k = 1000$  using greedy interpolation, as described in Section 6; the final point resolution is as in [Chen and Koltun 2015].

Figure 6 depicts the results of the DS++ algorithm and [Chen and Koltun 2015]. As can be read from the graphs, our algorithm compares favorably in both the inter and intra class matching scenarios in terms of cumulative error distribution and average error. These results are consistent for both the convex relaxation part (top row) and the upsampled final map (middle row); The graphs show our results both with our upsampling as described above (denoted by DS++(1)) and the results of combining our relaxation with the upsampling of [Chen and Koltun 2015] (DS++(2)). We find DS++(1) to be better on the inter class, and DS++(2) is marginally better on the intra class. The error is calculated on a set of 52 ground truth points in each mesh as in [Maron et al. 2016]. Figure 1 (left), and 3 show typical examples of maps computed using the DS++ algorithm in this experiment.

*Image arrangement.* The task of arranging image collections in a grid has received increasing attention in recent years [Carrizosa et al. 2016; Fried et al. 2015; Quadrianto et al. 2009; Strong and Gong 2014]. Image arrangement is an instance of metric matching: the first

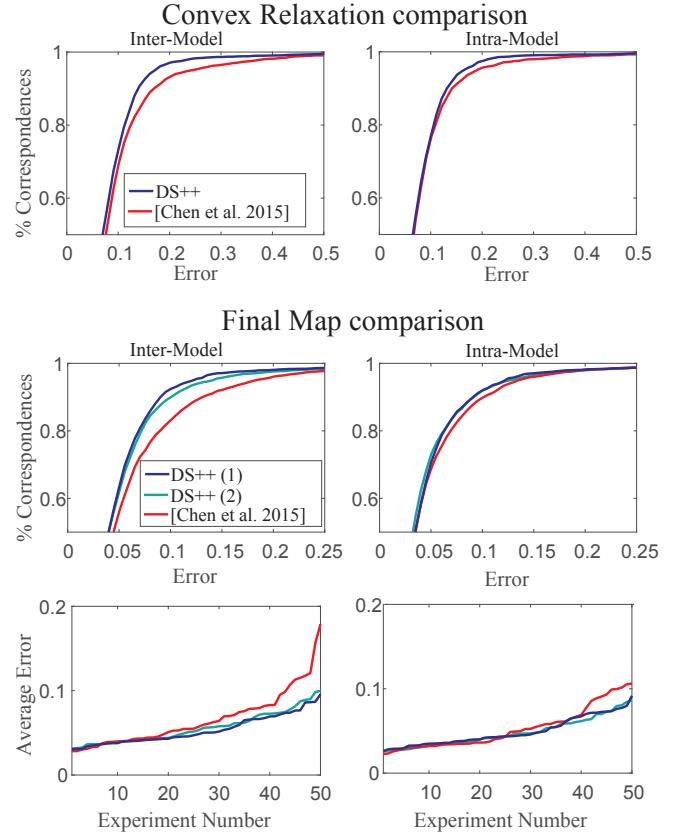


Fig. 6. Non-rigid matching. Cumulative and average errors achieved on the FAUST dataset [Bogo et al. 2014] by the DS++ algorithm compared to [Chen and Koltun 2015]. Top row compares only the convex relaxation part of both methods; bottom two rows compare final maps after upsampling. DS++(1) uses our upsampling method and DS++(2) uses the upsampling method of [Chen and Koltun 2015].

metric space is the collection of images and a dissimilarity measure defined between pairs of images; and the second, target metric space is a 2D grid (generally, a graph) with its natural Euclidean metric.

[Fried et al. 2015] suggested an energy functional for generating image arrangements, which are represented by a permutation matrix  $X$ . Their choice of energy functional was supported by a user study. This energy functional is:

$$E(X) = \min_{c>0} \sum_{ijkl} |c \cdot d_{ik} - d'_{jl}| X_{ij} X_{kl} \quad (8)$$

where  $d, d'$  are the distance measures between images and grid points respectively, and  $c$  is the unknown scale factor between the two metric spaces. [Fried et al. 2015] suggested a two step algorithm to approximate the minimizer of the above energy over the set of permutations: The first step is a dimensionality reduction, and the second is linear assignment to a grid according to Euclidean distances. Fried et al., demonstrated significant quantitative improvement over previous state of the art methods.

dataset	feature	improvement	rand average	Fried mean	our mean	functional	swaps?	grid size
<i>Random colors</i>	color	<b>28.33%</b>	0.478	0.259	<b>0.198</b>	Fried	no	12
<i>Random colors</i>	color	<b>8.86%</b>	0.478	0.219	<b>0.197</b>	Fried	yes	12
<i>Random colors</i>	color	<b>3.46%</b>	0.478	0.219	<b>0.211</b>	GW	yes	12
<i>SUN dataset</i>	color	<b>2.05%</b>	0.581	0.244	<b>0.237</b>	Fried	no	10
<i>SUN dataset</i>	color	<b>0.57%</b>	0.581	0.225	<b>0.223</b>	Fried	yes	10
<i>SUN dataset</i>	deep feature object	<b>55.97%</b>	0.433	0.345	<b>0.295</b>	Fried	no	14
<i>SUN dataset</i>	deep feature object	<b>6.31%</b>	0.433	0.300	<b>0.292</b>	Fried	yes	14
<i>LFW</i>	deep feature face	<b>50.70%</b>	0.422	0.355	<b>0.320</b>	Fried	no	14
<i>LFW</i>	deep feature face	<b>2.81%</b>	0.422	0.321	<b>0.318</b>	Fried	yes	14
<i>Illumination</i>	Raw $L_2$ distance	<b>59.08%</b>	0.509	0.320	<b>0.208</b>	Fried	no	10
<i>Illumination</i>	Raw $L_2$ distance	<b>9.94%</b>	0.509	0.232	<b>0.204</b>	Fried	yes	10
<i>Illumination</i>	Raw $L_2$ distance	<b>13.70%</b>	0.527	0.273	<b>0.238</b>	Fried	yes	10
<i>Illumination</i>	Raw $L_2$ distance	<b>10.65%</b>	0.518	0.259	<b>0.231</b>	Fried	yes	10

Table 1. Image arrangement comparison. We compare DS++ to [Fried et al. 2015] in arranging different sets of images in a grid with different affinity measures between images; see text for more details.

We perform image arrangement by using an alternative method for optimizing the energy (8). We fix  $c$  so that the mean of  $d$  and  $d'$  are the same, which leads to a quadratic matching energy which we optimize over permutations using the DS++ algorithm.

Table 1 summarizes quantitative comparison of the DS++ algorithm and [Fried et al. 2015] on a collection of different image sets and dissimilarity measures. Each row shows the mean energies over 100 experiments of Fried et al., DS++, and random assignments which provide a baseline for comparison; in each experiment we randomized a subset of images from the relevant set of images and generated image arrangements

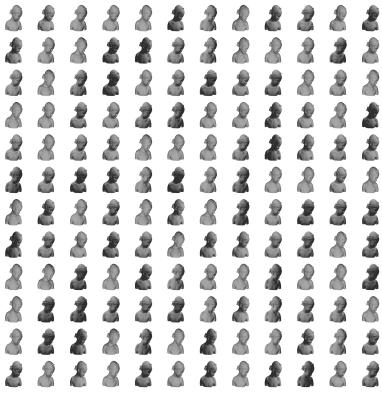


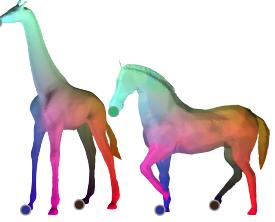
Fig. 7. Random lighting.

using the two methods. [Fried et al. 2015] suggested an optional post processing step in which random swaps are generated and applied in case they reduce the energy; our experiment measures the performance of both algorithms with and without random swaps. The first set of experiments tries to arrange random colors in a grid. The second set of experiments uses the mean image color for images from the SUN database [Xiao et al. 2010]. The third set uses the last layer of a deep neural network trained for object recognition [Chatfield et al. 2014] as image features, again for images in the SUN dataset. The fourth set of experiments organizes randomly sampled images from the Labeled Faces in the Wild (LFW) dataset [Huang et al. 2007] according to similar deep features taken from the net trained by [Parkhi et al. 2015]. For the last experiment, we rendered three 3D models from the SHREC07 dataset [Giorgi et al. 2007] from various illumination directions and

ordered them according to the raw  $L_2$  distance between pairs of images.

Our algorithm outperformed [Fried et al. 2015] in all experiments (in some cases our algorithm achieved an improvement of more than 50%). Figures 1, 4 and 8 show some image arrangements from these experiments. Note, for example, how similar faces are clustered together in Figure 8 (a), and similar objects are clustered in Figure 1 (right). Further note how the image arrangement in Figure 8 (b) nicely recovered the two dimensional parameter of the lighting direction, where Figure 7 shows the input random lighting directions renderings of a 3D models.

*Coarse-to-fine matching.* We consider the problem of matching two shapes  $\mathcal{S}$  and  $\mathcal{T}$  using sparse correspondences specified by the user. User input can be especially helpful for highly non-isometric matching problems where semantic knowledge is often necessary for achieving high quality correspondences. The inset shows such example where three points (indicated by colored circles) are used to infer correspondences between a horse and a giraffe.



We assume the user supplied a sparse set of point correspondences,  $s_i \rightarrow t_i$ ,  $i = 1, \dots, d$ , and the goal is to complete this set to a full correspondence set between the shapes  $\mathcal{S} = \{s_1, \dots, s_n\}$  and  $\mathcal{T} = \{t_1, \dots, t_k\}$ . Our general strategy is to use a linear term to enforce the user supplied constraints, and a quadratic term to encourage maps with low distortion.

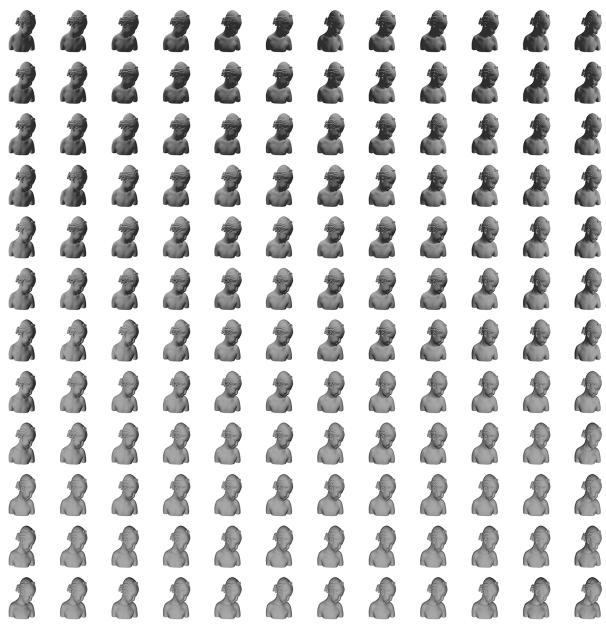
For a quadratic term we propose a "log-GW" functional. This functional amounts to choosing  $p(u, v) = d_L(u, v)^2$  for the definition of  $W$  in (2), where  $d_L$  is a metric on  $\mathbb{R}_+$  defined by

$$d_L(u, v) = \left| \log \frac{u}{v} \right|$$

This metric punishes for high relative distortion between  $u, v$ , and thus is more suitable for our cause than the standard Euclidean metric used for the GW functional.



(a)



(b)

Fig. 8. Generating automatic image arrangements with the DS++ algorithm. (a) Using deep features from a face recognition neural network cluster similar faces together, e.g., bald men (faces are picked at random from the LFW [Huang et al. 2007] dataset). (b) Automatic image arrangement of images of a 3D model with different lighting. (Images were randomly picked from a 30x30 noisy grid of illumination directions.) Note how the two dimensional lighting direction field is recovered by the DS++ algorithm: upper-right illuminated model image landed top-right in the grid, and similarly the other corners; images that are placed lower in the layout present more frontal illumination.

As a linear term we propose

$$L(X) = w \left[ \sum_{i=1}^d (-X_{ii}) + \sum_{i=1}^d \sum_{k,\ell} p(d(s_i, s_q), d(t_i, t_r)) X_{k\ell} \right]$$

The first summand from the left penalizes matchings which violate the known correspondences, while the second summand penalizes matchings which cause high distortion of distances to the user supplied points. The parameter  $w$  controls the strength of the linear term. In our experiments we chose  $w = 0.01 \|F^T WF\|$ , where  $\|F^T WF\| = \max\{|\bar{\lambda}_{\min}|, |\bar{\lambda}_{\max}|\}$  is the spectral norm of the quadratic form..

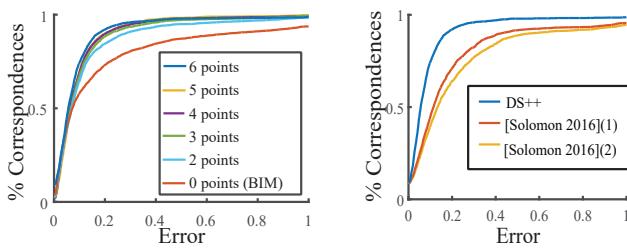


Fig. 9. Matching aided by sparse user correspondence. The left graphs shows that our algorithm can exploit user supplied information to outperform state of the art unsupervised methods such as BIM. The right graph shows DS++ outperforms the algorithm of [Solomon et al. 2016] for user aided matching.

We applied the algorithm for coarse-to-fine matching on the SHREC dataset [Giorgi et al. 2007], using  $d = 3, 4, 5, 6$  of the labeled ground truth points and evaluating the error of the obtained correspondence on the remaining  $\ell - d$  points. The number of labeled points  $\ell$  is class dependent and varies between 36 and 7.

Representative results are shown in Figure 10. The graph on the left hand side of Figure 9 shows that our algorithm is able to use this minimal user supplied information to obtain significantly better results than those obtained by the unaided BIM algorithm [Kim et al. 2011]. The graph compares the algorithms in terms of cumulative error distribution over all 218 SHREC pairs for which BIM results are available.

The graph on the right hand side of Figure 9 shows our results outperform the algorithm presented in [Solomon et al. 2016] for matching aided by user supplied correspondences. Both algorithms were supplied with 6 ground truth points. We ran the algorithm of [Solomon et al. 2016] matching  $n = 250, k = 250$  points (we take  $n = k$  since [Solomon et al. 2016] does not support injective matching) and using the maximal-coordinate projection they chose to achieve a permutation solution. These results are denoted by (2) in the graph. However we find that better results are achieved when matching only 100 points, and when using the  $L_2$  projection. These results are denoted by (1) in the graph.

**Timing.** Typical running times of our optimization algorithm for the energy of [Chen and Koltun 2015] matching  $n = k = 50$  points takes 6 seconds;  $n = k = 100$  takes 26 seconds; and  $n = 160, k = 150$  points takes around 2 minutes (130 seconds). The precomputation

of  $\bar{\lambda}_{\min}$  and  $\bar{\lambda}_{\max}$  with these parameters requires around 15 seconds, the  $L_2$  projection requires 5 seconds, and the remaining time is required for our optimization algorithm.

Parameter values of  $n = k = 100$  (as well as  $n = k = 12^2, 14^2$ ) were used in the image arrangement task from Section 8, and parameter values  $n = 160, k = 150$  were used for our results on the FAUST dataset. For the latter application we also upsampled to  $n = 450, k = 420$  using limited support interpolation and then upsampled to  $k = 1000, n = 5000$  using greedy interpolation as described in Section 6. Limited support interpolation required 117 seconds and greedy interpolation required 15 seconds. The total time for this application is around 4.5 minutes.

The efficiency of our algorithm significantly improves if the product of the quadratic form's matrix  $W$  with a vector  $x \in \mathbb{R}^{n^2}$  can be computed efficiently. This is illustrated by the fact that optimization of the sparse functional we construct for the task of resolution improvement with  $n = 450$  takes similar time as optimization of the non-sparse functional of [Chen and Koltun 2015] with  $n = 160$ .

Another case where the product  $Wx$  can be computed efficiently is the GW or log-GW energy. In both cases the product can be computed by multiplication of matrices of size  $n \times n$  (see [Solomon et al. 2016] for the derivation), thus using  $O(n^3)$  operations instead of the  $O(n^4)$  operations necessary for general  $W$ . Using this energy, matching  $n = 160$  points to  $k = 150$  points takes only 12 seconds, matching  $n = 270$  points to  $k = 250$  points takes 22 seconds, matching  $n = 500$  to  $k = 500$  takes 82 seconds, and for  $n = k = 1,000$  we require around six minutes (368 seconds).

The efficiency of our algorithm depends linearly on  $N$ . Minimizing the GW energy with  $n = 270, k = 250$  using  $N + 1 = 5$  sample takes 12 seconds, approximately half of the time needed when using  $N + 1 = 10$  samples. These parameters were used for our results on matching with user input described in Section 8. For this task we also used greedy interpolation to obtain full maps between the shapes, which required an additional 18 seconds. Overall this application required around half a minute.

Our algorithm was implemented on Matlab. All running times were computed using an Intel i7-3970X CPU 3.50 GHz.

## 9 CONCLUSIONS

We have introduced the DS++ algorithm for approximating the minimizer of a general quadratic energy over the set of permutations. Our algorithms contains two components: (i) A quadratic program convex relaxation that is guaranteed to be better than the prevalent doubly stochastic and spectral relaxations; and (ii) A projection procedure that continuously changes the energy to recover a locally optimal permutation, using the convex relaxation as an initialization. We have used recent progress in optimal transport to build an efficient implementation to the algorithm.

The main limitation of our algorithm is that it does not achieve the global minima of the energies optimized. Partially this is unavoidable and due to the computational hardness of our problem. However the experimental results in Figure 5 show that accuracy can be improved by the SDP method of [Kezurer et al. 2015] which while computationally demanding, can still be solved in polynomial time. Our future goal is to search for relaxations whose accuracy is close to

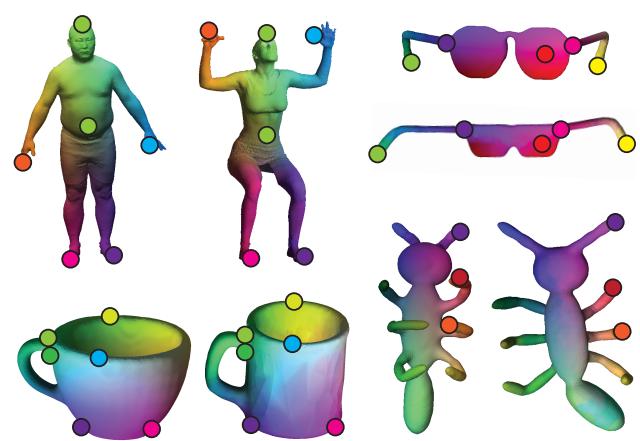


Fig. 10. Correspondences obtained using user input. Correspondences were obtained using 6 user input points, with the exception of the correspondence between ants found using only 3 points. Note our method is applicable to surfaces of arbitrary genus such as the genus 1 mugs.

those of [Kezurer et al. 2015] but which are also fairly scalable. One concrete direction of research could be finding the ‘best’ quadratic programming relaxation in  $O(n^2)$  variables.

## REFERENCES

- Warren P Adams and Terri A Johnson. 1994. Improved linear programming-based lower bounds for the quadratic assignment problem. *DIMACS series in discrete mathematics and theoretical computer science* 16 (1994), 43–75.
- Yonathan Aflalo, Alexander Bronstein, and Ron Kimmel. 2015. On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences* 112, 10 (10 March 2015), 2942–2947.
- Anonymous. 2017. DS++: A flexible, scalable, and provably tight relaxation for matching problems. (2017). arXiv:cs.GR/0902.0885
- Kurt M Anstreicher and Nathan W Brixius. 2001. A new bound for the quadratic assignment problem based on convex quadratic programming. *Mathematical Programming* 89, 3 (2001), 341–357.
- Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. 2015. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing* 37, 2 (2015), A1111–A1138.
- Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 3794–3801.
- Emilio Carrizosa, Vanesa Guerrero, and Dolores Romero Morales. 2016. Visualizing proportions and dissimilarities by Space-filling maps: a Large Neighborhood Search approach. *Computers & Operations Research* (2016).
- Ken Chatfield, Karen Simonyan, Andrej Vedaldi, and Andrew Zisserman. 2014. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531* (2014).
- Qifeng Chen and Vladlen Koltun. 2015. Robust Nonrigid Registration by Convex Optimization. In *Proceedings of the IEEE International Conference on Computer Vision*. 2039–2047.
- Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*. 2292–2300.
- Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. 2008. Performance capture from sparse multi-view video. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 98.
- Yichuan Ding and Henry Wolkowicz. 2009. A low-dimensional semidefinite relaxation for the quadratic assignment problem. *Mathematics of Operations Research* 34, 4 (2009), 1008–1022.
- Nadav Dym and Yaron Lipman. 2016. Exact Recovery with Symmetries for Procrustes Matching. *arXiv preprint arXiv:1606.01548* (2016).
- Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. 1997. The farthest point strategy for progressive image sampling. *Image Processing, IEEE Transactions on* 6, 9 (1997), 1305–1315.

- 1 Wei Feng, Jin Huang, Tao Ju, and Hujun Bao. 2013. Feature correspondences using  
2 Morse smale complex. *The Visual Computer* 29, 1 (2013), 53–67.
- 3 Marcelo Fiori and Guillermo Sapiro. 2015. On spectral properties for graph matching  
4 and graph isomorphism problems. *Information and Inference* 4, 1 (2015), 63–76.
- 5 Fajwel Fogel, Rodolphe Jenatton, Francis Bach, and Alexandre d’Aspremont. 2013.  
Convex relaxations for permutation problems. In *Advances in Neural Information  
Processing Systems*. 1016–1024.
- 6 F. Fogel, R. Jenatton, F. Bach, and A. d’Aspremont. 2015. Convex Relaxations for  
7 Permutation Problems. *SIAM J. Matrix Anal. Appl.* 36, 4 (2015), 1465–1488.
- 8 Ohad Fried, Stephen DiVerdi, Maciej Halber, Elena Sizikova, and Adam Finkelstein.  
2015. IsoMatch: Creating informative grid layouts. In *Computer Graphics Forum*,  
Vol. 34. Wiley Online Library, 155–166.
- 9 Andrew H Gee and Richard W Prager. 1994. Polyhedral combinatorics and neural  
networks. *Neural computation* 6, 1 (1994), 161–180.
- 10 Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. 2007. Shape retrieval contest  
11 2007: Watertight models track. *SHREC competition* 8 (2007).
- 12 Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. 2007. *Labeled faces  
in the wild: A database for studying face recognition in unconstrained environments.*  
Technical Report. Technical Report 07-49, University of Massachusetts, Amherst.
- 13 Itay Kezurer, Shahar Z. Kovalsky, Ronen Basri, and Yaron Lipman. 2015. Tight Relaxation  
14 of Quadratic Matching. *Comput. Graph. Forum* 34, 5 (Aug. 2015), 115–128.
- 15 Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. 2011. Blended intrinsic maps.  
In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 79.
- 16 Vladimir Kolmogorov. 2006. Convergent tree-reweighted message passing for energy  
minimization. *IEEE transactions on pattern analysis and machine intelligence* 28, 10  
(2006), 1568–1583.
- 17 JJ Kosowsky and Alan L Yuille. 1994. The invisible hand algorithm: Solving the assign-  
ment problem with statistical physics. *Neural networks* 7, 3 (1994), 477–490.
- 18 Marius Leordeanu and Martial Hebert. 2005. A spectral technique for correspondence  
problems using pairwise constraints. In *Tenth IEEE International Conference on  
Computer Vision (ICCV’05) Volume 1*, Vol. 2. IEEE, 1482–1489.
- 19 Yaron Lipman and Thomas Funkhouser. 2009. Möbius voting for surface correspondence.  
In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 72.
- 20 Jingjin Liu, Jiebo Luo, and Mubarak Shah. 2009. Recognizing realistic actions from  
videos à Äjin the wild à Ä. In *Computer Vision and Pattern Recognition, 2009. CVPR  
2009. IEEE Conference on*. IEEE, 1996–2003.
- 21 Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter  
Hahn, and Tania Querido. 2007. A survey for the quadratic assignment problem.  
*European journal of operational research* 176, 2 (2007), 657–690.
- 22 Vince Lyzinski, Donniell E. Fishkind, Marcelo Fiori, Joshua T. Vogelstein, Carey E.  
Priebe, and Guillermo Sapiro. 2016. Graph Matching: Relax at Your Own Risk. *IEEE  
Trans. Pattern Anal. Mach. Intell.* 38, 1 (2016), 60–73.
- 23 Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. 2016.  
Point Registration via Efficient Convex Relaxation. *ACM Trans. Graph.* 35, 4, Article  
73 (July 2016), 12 pages.
- 24 Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. 2015.  
Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings  
of the IEEE international conference on computer vision workshops*. 37–45.
- 25 Facundo Mémoli. 2011. Gromov-Wasserstein distances and the metric approach to  
object matching. *Foundations of computational mathematics* 11, 4 (2011), 417–487.
- 26 Facundo Mémoli and Guillermo Sapiro. 2005. A theoretical and computational frame-  
work for isometry invariant recognition of point cloud data. *Foundations of Computational  
Mathematics* 5, 3 (2005), 313–347.
- 27 Richard G Ogier and DA Beyer. 1990. Neural network solution to the link scheduling  
problem using convex relaxation. In *Global Telecommunications Conference, 1990,  
and Exhibition.‘Communications: Connecting the Future’, GLOBECOM’90*. IEEE, IEEE,  
1371–1376.
- 28 Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas  
Guibas. 2012. Functional maps: a flexible representation of maps between shapes.  
*ACM Transactions on Graphics (TOG)* 31, 4 (2012), 30.
- 29 Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep face recognition.  
In *British Machine Vision Conference*, Vol. 1. 6.
- 30 Novi Quadrianto, Le Song, and Alex J Smola. 2009. Kernelized sorting. In *Advances in  
neural information processing systems*. 1289–1296.
- 31 Anand Rangarajan, Steven Gold, and Eric Mjolsness. 1996. A novel optimizing network  
architecture with applications. *Neural Computation* 8, 5 (1996), 1041–1060.
- 32 Anand Rangarajan, Alan L Yuille, Steven Gold, and Eric Mjolsness. 1997. A conver-  
gence proof for the softassign quadratic assignment algorithm. *Advances in neural  
information processing systems* (1997), 620–626.
- 33 Emanuele Rodola, Alex M Bronstein, Andrea Albarelli, Filippo Bergamasco, and An-  
drea Torsello. 2012. A game-theoretic approach to deformable shape matching. In  
*Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE,  
182–189.
- 34 Emanuele Rodolà, Samuel Rota Bulò, Thomas Windheuser, Matthias Vestner, and Daniel  
Cremers. 2014. Dense non-rigid shape correspondence using random forests. In  
*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- 35 Christian Schellewald, Stefan Roth, and Christoph Schnörr. 2001. Evaluation of convex  
optimization techniques for the weighted graph-matching problem in computer  
vision. In *Joint Pattern Recognition Symposium*. Springer, 361–368.
- 36 Tianjia Shao, Wilmot Li, Kun Zhou, Weiwei Xu, Baining Guo, and Niloy J Mitra. 2013.  
Interpreting concept sketches. *ACM Transactions on Graphics (TOG)* 32, 4 (2013),  
56.
- 37 Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher,  
Andy Nguyen, Tao Du, and Leonidas Guibas. 2015. Convolutional wasserstein  
distances: Efficient optimal transportation on geometric domains. *ACM Transactions  
on Graphics (TOG)* 34, 4 (2015), 66.
- 38 Justin Solomon, Andy Nguyen, Adrian Butscher, Mirela Ben-Chen, and Leonidas Guibas.  
2012. Soft maps between surfaces. In *Computer Graphics Forum*, Vol. 31. Wiley Online  
Library, 1617–1626.
- 39 Justin Solomon, Gabriel Peyré, Vladimir G. Kim, and Suvrit Sra. 2016. Entropic Metric  
Alignment for Correspondence Problems. *ACM Trans. Graph.* 35, 4, Article 72 (July  
2016), 13 pages.
- 40 Grant Strong and Minglun Gong. 2014. Self-sorting map: An efficient algorithm for  
presenting multimedia data in structured layouts. *IEEE Transactions on Multimedia*  
16, 4 (2014), 1045–1058.
- 41 Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. 2011. A survey  
on shape correspondence. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library,  
1681–1707.
- 42 Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. 2016. Dense  
Human Body Correspondences Using Convolutional Networks. In *Computer Vision  
and Pattern Recognition (CVPR)*.
- 43 Tomas Werner. 2007. A linear programming approach to max-sum problem: A review.  
*IEEE transactions on pattern analysis and machine intelligence* 29, 7 (2007).
- 44 Yong Xia. 2008. Second order cone programming relaxation for quadratic assignment  
problems. *Optimization Methods & Software* 23, 3 (2008), 441–449.
- 45 Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010.  
Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision  
and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, 3485–3492.
- 46 Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. 2009. A path following  
algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis  
and Machine Intelligence* 31, 12 (2009), 2227–2242.
- 47 Yun Zeng, Chaohui Wang, Yang Wang, Xianfeng Gu, Dimitris Samaras, and Nikos Paragios.  
2010. Dense non-rigid surface registration using high-order graph matching.  
In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE,  
382–389.
- 48 Qing Zhao, Stefan E Karisch, Franz Rendl, and Henry Wolkowicz. 1998. Semidefinite  
programming relaxations for the quadratic assignment problem. *Journal of  
Combinatorial Optimization* 2, 1 (1998), 71–109.
- 49 Silvia Zuffi and Michael J Black. 2015. The Stitched Puppet: A Graphical Model of 3D  
Human Shape and Pose. In *Proceedings of the IEEE Conference on Computer Vision  
and Pattern Recognition*. 3537–3546.
- 50 **A PROOFS**
- 51 *Proof of Lemma 3.1.* The function  $f(X) = \|X\|_F^2$  is strictly convex  
and satisfies  $f(X) = n$  for all extreme points of  $DS$ . Therefore
- 52 
$$E(X, b) - E(X, a) = (a - b) \|X\|_F^2 + (b - a)n \geq 0$$
- 53 with equality iff  $X$  is a permutation.
- 54 *Proof of Lemma 4.2.* We omit the proof of the first part of the  
lemma since it is similar to, and somewhat easier than, the proof of  
the second part.
- 55 To show equivalence of  $DS++$  with (6) we show that every mini-  
mizer of  $DS++$  defines a feasible point for (6) with equal energy and  
vice versa.
- 56 Let  $x$  be the minimizer of  $E(X, \bar{\lambda}_{\min})$  over the doubly stochastic  
matrices and let  $v$  be the eigenvector of  $F^T WF$  of unit Euclidean  
norm corresponding to its minimal eigenvalue  $\bar{\lambda}_{\min}$ . Denote  $u = Fv$ .  
We define  $Y = xx^T + \alpha uu^T$ , where we choose  $\alpha \geq 0$  so that (6b)  
holds. This is possible since  $\text{tr}(xx^T) = \|X\|_F^2 \leq n$ . Further note that  
 $Y$  also satisfies (6f) since  $\alpha \geq 0$ , and (6e) since
- 57 
$$AY = Ax x^T + \alpha A u u^T = bx^T + \alpha A(Fv)(Fv)^T = bx^T$$

where we used the fact that  $Fv$  is a solution to the homogeneous linear equation  $Ax = 0$ . Finally the energy satisfies (ignoring the constant  $d$ )

$$\begin{aligned} E(X, Y) &= \text{tr}WY + c^T x \\ &= x^T Wx + \alpha v^T F^T WFv + c^T x \\ &= x^T Wx + \alpha \bar{\lambda}_{\min} + c^T x \\ &= x^T Wx + (n - \|X\|_F^2) \bar{\lambda}_{\min} + c^T x \\ &= E(X, \bar{\lambda}_{\min}) \end{aligned}$$

Now let  $(X, Y)$  be a minimizer of (6), we show that  $x$  is a feasible solution of our relaxation with the same energy. In fact due to the previous claim it is sufficient to show that  $E(X, \bar{\lambda}_{\min}) \leq E(X, Y)$ . The feasibility of  $X$  is clear since it is already DS. Next, denote

$$W_{\lambda} = W - \bar{\lambda}_{\min} I$$

then (ignoring the constant  $d$ )

$$\begin{aligned} E(X, \bar{\lambda}_{\min}) &= \text{tr}W_{\lambda}xx^T + c^T x + \bar{\lambda}_{\min}n \\ &\stackrel{(*)}{\leq} \text{tr}W_{\lambda}Y + c^T x + \bar{\lambda}_{\min}n \\ &\stackrel{(6b)}{=} \text{tr}WY + c^T x = E(X, Y) \end{aligned}$$

The inequality  $(*)$  follows from the fact that  $A(Y - xx^T) = 0$  due to (6d),(6e) and therefore since  $FF^T$  is the projection onto the kernel of  $A$ :

$$FF^T(Y - xx^T) = Y - xx^T = (Y - xx^T)FF^T$$

and so  $(*)$  follows from

$$\begin{aligned} \text{tr}W_{\lambda}(Y - xx^T) &= \text{tr}W_{\lambda}FF^T(Y - xx^T)FF^T \\ \text{tr}[F^T W_{\lambda} F][F^T (Y - xx^T) F] &\geq 0 \end{aligned}$$

where the last inequality follows from the fact that the two matrices in square brackets are positive semi-definite due to the definition of  $\bar{\lambda}_{\min}$  and (6f).

We now prove the third part of the lemma:

*Comparison with SDP relaxation.* The SDP relaxation of [Kezurer et al. 2015] is

$$\max_Y \text{tr}WY + c^T x + d \quad (9a)$$

$$\text{s.t. } \text{tr}Y = n \quad (9b)$$

$$x \geq 0 \quad (9c)$$

$$Ax = b \quad (9d)$$

$$Y \geq xx^T \quad (9e)$$

$$Y \geq 0 \quad (9f)$$

$$\sum_{qrst} Y_{qrst} = n^2 \quad (9g)$$

$$Y_{qrst} \leq \begin{cases} 0, & \text{if } q = s, r \neq t \\ 0, & \text{if } r = t, q \neq s \\ \min \{X_{qr}, X_{st}\}, & \text{otherwise} \end{cases} \quad (9h)$$

where  $Y_{qrst}$  is the entry replacing the quadratic monomial  $X_{qr}X_{st}$ . We note this relaxation contains all constraints from the SDP relaxation (6) with the exception of (6e). It also contains the additional

constraints (9f)-(9h) which do not appear in (6). Thus to show that [Kezurer et al. 2015] is tighter than our relaxation it is sufficient to show that (6e) is implied by the other constraints of [Kezurer et al. 2015]. We recall that (6e) represent all constraints obtained by multiplying linear equality constraints by a linear monomial.

For a quadratic polynomial

$$g(x) = x^T Wx + c^T x + e$$

let us denote by  $\bar{g}(x, Y)$  the linearized polynomial

$$\bar{g}(x, Y) = \text{tr}WY + c^T x + e$$

We will use the following property of SDP relaxations (see [Dym and Lipman 2016]): If a quadratic polynomial  $g$  is of the form  $g = p^2$  then

(1) For any feasible  $x, Y$  we have  $\bar{g}(x, Y) \geq 0$ .

(2) If  $\bar{g}(x, Y) = 0$  is satisfied for all feasible  $x, Y$ , then for any quadratic  $f$  of the form  $f = pq$  we have  $\bar{f}(x, Y) = 0$ .

Accordingly, it is sufficient to show that the squares  $g_q = p_q^2, h_r = m_r^2$  of all the linear equality polynomials

$$p_q(X) = \sum_r X_{qr} - 1, \quad m_r(X) = \sum_r X_{qr} - 1$$

satisfy  $\bar{g}_q = 0, \bar{h}_r = 0$ . We obtain  $\bar{g}_q = 0$  from

$$\begin{aligned} 0 \leq \bar{g}_q(x, Y) &= \sum_{r,t} Y_{qrqt} - 2 \sum_r X_{qr} + 1 \\ &\stackrel{(9h)}{=} \sum_r X_{qr} - 2 \sum_r X_{qr} + 1 = 0 \end{aligned}$$

the proof that  $\bar{h}_r = 0$  is identical.

## B SPARSITY PATTERN FOR IMPROVING MATCHING RESOLUTION

We construct a sparsity pattern for the task of matching  $s_1, \dots, s_k$  to  $t_1, \dots, t_n$  using known correspondences  $\hat{s}_\ell \mapsto \hat{t}_\ell, \ell = 1, \dots, r$ .

For each  $s_i$  we use the following procedure to determine which correspondence will be forbidden: We find the five matched points  $\hat{s}_{\ell_1}, \dots, \hat{s}_{\ell_5}$  which are closest to  $s_i$  and compute the geodesic distance of these points from  $s_i$ . This gives us a feature vector  $v \in \mathbb{R}^5$ . We then compute the geodesic distances of each of the points  $t_j, j = 1, \dots, n$  from the matched points  $\hat{t}_{\ell_1}, \dots, \hat{t}_{\ell_5}$  corresponding to the five closest points to  $s_i$ . This gives us  $n$  feature vectors  $v_j \in \mathbb{R}^5$ . For  $t_j$  to be a viable match we require that  $\|v_j - v\|_2$  be small. We therefore allow the top 20% of the correspondences according to this criteria.

To symmetrize this process, we use the same procedure to find permissible matches for each  $t_j$ , and then select as permissible all matches  $s_i \mapsto t_j$  which were found permissible either when starting from  $s_i$  or when starting from  $t_j$ .