Efficient high order matching

Michael Chertok and Yosi Keller,

School of Engineering, Bar-Ilan University, Israel.

Email: yosi.keller@gmail.com

Abstract

We present a computational approach to high order matching of datasets in $\mathbb{R}^d$. Those are matchings based on data affinity measures that score the matching of more than two pairs of points at a time. High order affinities are represented by tensors and the matching is then given by a rank-one approximation of the affinity tensor and a corresponding discretization. Our approach is rigorously justified by extending Zass and Shashua's hypergraph matching [40] to high order spectral matching. This paves the way for a computationally efficient dual marginalization-spectral matching scheme. We also show that based on the spectral properties of random matrices, affinity tensors can be randomly sparsified while retaining the matching accuracy. Our contributions are experimentally validated by applying them to synthetic as well as real datasets.

1 Introduction

Graph matching is a fundamental problem in theoretical computer science, and relates to a gamut of research topics in computer vision and machine learning. Those include the registration of sets of points in $\mathbb{R}^d$ [22, 38], object recognition [6], shape retrieval [12] and symmetry analysis [16], to name a few. Given a set of points, graph nodes can represent the points and the graph edges encode their distances. Hence, computing graph correspondence paves the way for recovering point correspondences.

Over the years, a myriad of point matching approaches were suggested. These can be categorized as those that solve the linear assignment problem (bipartite graph matching) Vs. those that solve high order assignment (HOA) problems. Given two sets of points in $\mathbb{R}^d$, such that $S_1 = \{\mathbf{x}_i\}_{i=1}^{n_1}$ and $S_2 = \{\mathbf{y}_i\}_{i=1}^{n_2}$, the assignment problem is to find the set of assignments

$$C \triangleq \{c_i\}_{i=1}^{n_1} = \{\mathbf{x}_i, \mathbf{y}_{i'}\}_{i=1}^{n}. \tag{1.1}$$

WLOG it is common to add a dummy element $\mathbf{y}_0$ to the set S_2 such that $\{\mathbf{x}_i, \mathbf{y}_0\}$ implies that $\mathbf{x}_i$ is not matched to any point in S_2. We represent the correspondence by an indicator vector $\mathbf{z} \in \{0,1\}^{n_1 n_2}$, which is the row-wise vectorized replica of the assignment matrix $Z \in \{0,1\}^{n_1 \times n_2}$, where $Z_{ii'} = 1$ implies that $\mathbf{x}_i$ corresponds to $\mathbf{y}_{i'}$, and $Z_{ii'} = 0 \; \forall i'$ implies that $\mathbf{x}_i$ is not matched to any point in S_2.

The matching is defined by an affinity matrix A such that

$$A_{i,i'} = \Omega_1 (\mathbf{x}_i, \mathbf{y}_{i'}) = \Omega_1 (c_i),$$

is the affinity of matching $\mathbf{x}_i$ to $\mathbf{y}_{i'}$, and Ω_1 is the linear assignment affinity measure. The optimal assignment is thus given by

$$\mathbf{z}^* = \arg\max_{\mathbf{z}} \left(\mathbf{z}^T \mathbf{a}\right), \mathbf{z}^* \in \{0,1\}^{n_1 n_2}$$

$$s.t.\; Z^* \underline{\mathbf{1}} \leqslant \underline{\mathbf{1}} \text{ and } \left(Z^*\right)^T \underline{\mathbf{1}} \leqslant \underline{\mathbf{1}} \tag{1.2}$$

where $\mathbf{a} \in \mathbb{R}^{n_1 n_2}$ is a row-wise vectorized replica of A, and $Z^* \in \{0,1\}^{n_1 \times n_2}$ is a row-wise reshaped replica of $\mathbf{z}^*$.

The constraints in Eq. 1.2 imply that a point $\mathbf{x}_i \in S_1$ can only be matched to a single point in S_2, or not matched at all. The same applies to each point $\mathbf{y}_i \in S_2$. Equation 1.2 can be optimally solved by the Hungarian algorithm [31] or approximated by Dynamic Programming [5].

In pairwise assignments we are given a pairwise affinity measure Ω_2 that scores the simultaneous matching of c_i and c_j

$$\Omega_2 = \Omega_2 (c_i, c_j),$$

implying that $\mathbf{x}_i$ is matched to $\mathbf{y}_{i'}$ and $\mathbf{x}_j$ to $\mathbf{y}_{j'}$ simultaneously. Pairwise affinities can encode geometric properties such as *local isometry*

$$\begin{aligned}\Omega_2 (c_i, c_j) &= \Omega_2 \left(\{\mathbf{x}_i, \mathbf{y}_{i'}\}, \{\mathbf{x}_j, \mathbf{y}_{j'}\}\right) \\ &= \exp -\frac{1}{\varepsilon^2} \left(\|\mathbf{x}_i - \mathbf{x}_j\|_2 - \left\|\mathbf{y}_{i'} - \mathbf{y}_{j'}\right\|_2\right)^2,\end{aligned} \tag{1.3}$$

where $\varepsilon > 0$ is the kernel bandwidth. The affinity matrix A is given by

$$A\left(i (n_2 - 1) + i', j (n_2 - 1) + j'\right) = \Omega_2 (c_i, c_j),$$

and the optimal assignment is given by the one maximizing the sum of corresponding pairwise affinities

$$C^{*} = \max_{C} \arg \sum_{C} \Omega_{2}\left(c_{i}, c_{j}\right)$$

adhering to the same set of constraints as in Eq. 1.2.

This yields the following binary optimization problem

$$\mathbf{z}^{*} = \arg\max_{\mathbf{z}} \left(\mathbf{z}^{T} A \mathbf{z}\right), \mathbf{z}^{*} \in \{0,1\}^{n_1 n_2}$$
$$s.t.\ Z^{*}\underline{\mathbf{1}} \leqslant \underline{\mathbf{1}} \text{ and } \left(Z^{*}\right)^{T} \underline{\mathbf{1}} \leqslant \underline{\mathbf{1}} \tag{1.4}$$

where $\mathbf{z}$ and Z are the same indicator vector/matrix pair as in Eq. 1.2, and $A \in \mathbb{R}^{n_1 n_2 \times n_1 n_2}$. The binary quadratic optimization problem in Eq. 1.4 is np-complete. Leordeanu and Herbert [22] showed how to apply spectral relaxation to the pairwise HOA and derived an efficient and robust computational solution by way of spectral relaxation

$$\mathbf{w}^{*} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^{T} A \mathbf{w}}{\mathbf{w}^{T} \mathbf{w}}, \quad \mathbf{w} \in \mathbb{R}^{n_1 n_2}. \tag{1.5}$$

Equation 1.5 is solved by computing the leading eigenvalue and corresponding eigenvector of A. In point matching it is common to compute the affinity using a Gaussian kernel as in Eq. 1.3, which guarantees that A is symmetric and nonnegative. Thus, by the Perron-Frobenius theorem we have that the leading eigenvalue and eigenvector $\mathbf{w}^{*}$ of A are known to exist and $\mathbf{w}^{*}$ is nonnegative. Given the continuous solution $\mathbf{w}^{*}$, Leordeanu *et al.* [22] suggest to discretize $\mathbf{w}^{*}$ by a greedy approach and derive the indicator vector $\mathbf{z}^{*}$. The assignment constraints in Eq. 1.4 are ignored in the spectral relaxation step (Eq. 1.5) and induced during the discretization step.

Partial matching implies that only subsets of size $n \leqslant \min\left(n_1, n_2\right)$ of S_1 and S_2 can be matched. This is of particular interest in applications where the common set is embedded in clutter. In applications such as image registration, it is possible to restrict the space of potential assignments of each point $\mathbf{x}_i \in S_1$, by using *local descriptors* [29]. Those characterize each point and allow to search for its $K \ll n_2$ nearest neighbors in S_2. This reduces the dimensionality of the terms in Eq. 1.5 to $A \in \mathbb{R}^{n_1 K \times n_1 K}$ and $\mathbf{w}^{*} \in \mathbb{R}^{n_1 K}$.

Although pairwise matching was found to be instrumental [6, 16, 13], some applications require high order affinities. For instance, consider the point sets depicted in Fig. 1. The local isometric matching in Eq. 1.3 will fail to register these images, as it is susceptible to the scale differences between the sets of

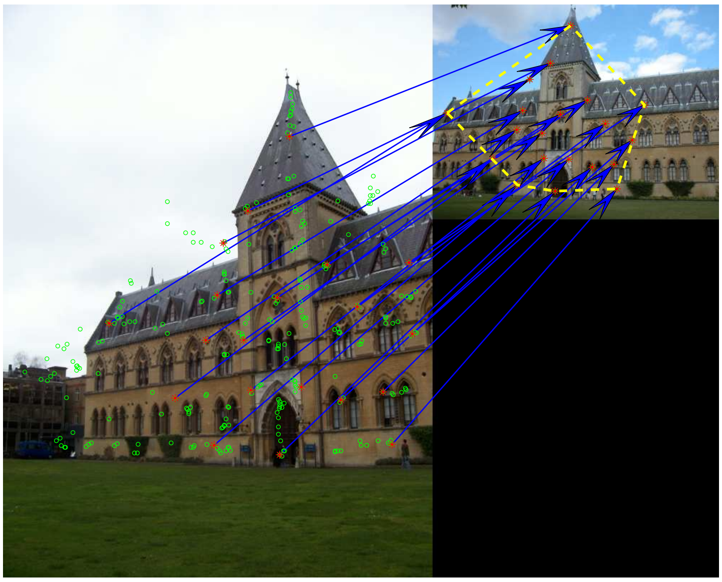


Figure 1: Image registration by high order matching. The images were taken from the Oxford Buildings dataset [32]. The right image was downscaled by a factor of two. Due to the scale differential the images can not be matched by pairwise affinity measures.

points in each image. Using triplets affinities as in Fig. 2, we can match triangles and thus define a scale invariant similarity measure. This is the core motivation of our work.

In this work we present and rigorously justify a computational approach to m-order high order matching. As m-order affinities are represented by m-order tensors we denote our scheme Tensor-HOA. In particular, we present the following contributions:

First, we extend the probabilistic interpretation of hypergraph matching by Zass and Shashua [40] to derive a novel *spectral* high order matching scheme that is based on rank-one approximation (ROA) of tensors, by way of tensor eigendecomposition. We discuss the numerical attributes and downsides of such decompositions and propose a dual marginalization-decomposition scheme that reduces a m-order ROA problem to a second order ROA problem that can be solved efficiently and optimally by matrix eigendecomposition. This improves the matching accuracy and robustness and allows to reduce the memory storage required for the HOA, as that is a fundamental obstacle to the feasibility and practical use of high order matching schemes.

Second, we utilize recent results in random matrix theory to sparsify the affinity tensor T_m. Thus, we are able to accurately compute the ROA of T_m while computing a small random subset of its elements.

We show this scheme to be particularly appropriate for sparse tensor such as the affinity tensor used by the HOA.

Last, we show that the discretization step of the relaxation is essentially the solution of a linear assignment problem and study different approaches for its computation.

This paper is organized as follows: Section 2 discusses previous results on graph matching and presents the Tensor-HOA scheme in Section 3. A probabilistic justification of the Tensor-HOA scheme is presented in Section 3.2, which paves the way for a dual marginalization-spectral matching scheme, that improves on the Tensor-HOA. The Sparse Tensor-HOA is discussed in Section 3.3. Our approach is experimentally verified in Section 4 by applying it to synthetic as well as real data. Concluding remarks and future extensions are discussed in Section 5.

2 Related work

A myriad of algorithms were proposed for matching graphs and sets of point in $\mathbb{R}^d$, and a comprehensive survey was given by Conte *et al.* [9]. The first class of algorithm solves the bipartite graph matching problem, or equivalently the LAP, as in Eq. 1.2. One of the first such algorithms for point matching was proposed by Scott and Longuet-Higgins [35]. Their method solves the one-to-one matching by minimizing the sum of squared distances between matched points in both sets. It introduces a spectral formulation by computing a Gaussian-weighted affinity matrix $A_{i,j} = \exp - \|\mathbf{x}_i - \mathbf{y}_j\|^2 / \varepsilon^2$. A is replaced by $P = UV$, where U and V are matrices whose columns are the left and right singular vectors of A. The Largest entries of P indicate strongly coupled points. This scheme implicitly assumes that A is an empirical approximation of the one-to-one assignment matrix $\widehat{A}$, for which we have $\widehat{A}\underline{\mathbf{1}}=\widehat{A}^T\underline{\mathbf{1}}=\underline{\mathbf{1}}$. It was observed by Cour *et al.* [10] that this spectral decomposition acts as a projection operator into the space of assignment matrices.

The Graduated Assignment algorithm of Gold and Rangarajan [13] was one of the first to solve the quadratic assignment problem (QAP). By relaxing the constraint $\mathbf{z}^* \in \{0,1\}^{n_1 n_2}$ in Eq. 1.4 to $\mathbf{w} \in \mathbb{R}^{n_1 n_2}$. The derivative of the relaxed QAP is computed and an iterative update scheme is derived. The relaxation in this scheme boils down to a power iteration and thus precedes [22]. In addition, the authors introduce a stretching operator that iteratively increases high soft assignment values, and decreases lower ones. The stretching is controlled by a continuation parameter, annealed after each iteration. In Section 3.2 we show this scheme to be an Estimate Maximize (EM) estimate of the soft assignments.

Maciel and Costeira [27] formulated pairwise matching as a constrained integer optimization problem with a concave optimization scheme, whose complexity is not polynomial, and thus computationally slow. Cour *et al.* [10] introduced two valuable extensions to Leordeanu's work. First, they introduce affine constraints into the spectral decomposition that encode the one-to-one matching constraints. In Leordeanu's work, these constraints were overlooked at the spectral decomposition step, and imposed in the discretization step. Their second contribution is to apply bistochastic normalization to the Edge Similarity Matrix, which is a an equivalent representation of the pairwise affinity matrix.

Torresani *et al.* [38] proposed a graph matching scheme they denote as *dual decomposition* for solving the QAP. They use the Nonlinear Inverse Optimization algorithm [25] to self-tune the parameters of the matching scheme. The approach was shown to be more robust than spectral relaxation, but slower by two orders of magnitude.

The work of Zass and Shashua [40] inspired our probabilistic analysis. They were the first to present a probabilistic framework for hypergraph matching, by assuming that different assignments are statistically independent. This builds upon their previous results on the connection between low-rank tensors and conditional independence [36]. The high order formulation represented by tensors is shown to encode high order geometric invariants such as scale and affine invariants. Zass and Shashua also present two contributions that constitute a high order graph matching algorithm: first they marginalize the high order tensor into a one-dimensional probability vector, thus avoiding the need for high order rank-one-approximation. This probability vector is refined by projecting it into the space of assignment vectors by minimizing a Bergman measure.

In this work we extend their results to derive a *spectral* approach to high order matching. By analyzing the downsides of high order eigendecompositions we suggest a dual marginalization-spectral approach that outperforms the pure spectral scheme.

The work of Duchenne *et al.* [11] is of particular interest as they *independently* presented a spectral high order scheme similar to ours. They extend Leordeanu's *et al.* approach [22] by replacing the affinity matrix with an affinity tensor and using the High Order Power Iteration. They describe the construction of affinity tensors that can be made scale, affinity and projective-invariant. To deal with the increased computational complexity and storage, they propose to only consider triangles (for third-order affinities) that consist of close points in the source image (S_1) and compute all of the triangles in the target image (S_2).

High order matching was applied by Mammalian *et al.* [30] to matching corresponding moles in patient's skin at different scanning times. In order to measure the distances between moles in geodesic coordinates on the body surface, the image of the back is segmented and fitted to a surface model. The matching is then formulated by high order graph matching where the affinity tensor consists of a combination of geometric and anatomical terms. This work exemplifies the applicability and flexibility of graph matching.

Hypergraphs were shown to be instrumental in high order clustering and classification, where the relationships between the objects of interest are more complex than pairwise. Zhou *et al.* [41] proposed to use hypergraphs to represent such complex relationships and derived a spectral clustering approach for hypergraphs and transductive classification. They present the equivalent of a Normalized Cut on hypergraphs and a random walk interpretation, similar to previous results in pairwise clustering [37].

Govindu [15] and Agarwal *et al.* [3] proposed to explicitly represent high order geometric affinities by an affinity tensor. The affinities are the fitting error of a parametric geometrical models such as a conic or an affine motion. In [15], the affinity tensor is decomposed by first flattening it, as in the High Order SVD, and utilizing its supersymmetry to randomly subsample the flattened tensor and compute its eigenvectors using a subset of matrix. In [3] the hypergraph is reduced to a graph by Clique Averaging, and is partitioned by spectral clustering.

Theoretical analysis of the graph Laplacians of hypergraphs was conducted by Agarwal *et al.* [2]. They analyze the structure of hypergraphs for unsupervised and semi-supervised learning and show how to reduce the hypergraphs to pairwise graphs represented by affinity matrices and their corresponding graph Laplacians.

Note that these works [41, 15, 3, 2] deal with classification and clustering while our work solves the graph matching problem.

3 High Order Assignment

High order assignment is an extension of the pairwise formulation [22]. Given two sets of points in $\mathbb{R}^d$, such that $S_1 = \{\mathbf{x}_i\}_{i=1}^{n1}$ and $S_2 = \{\mathbf{y}_i\}_{i=1}^{n2}$, the m-order affinity measure Ω_m quantifies the matching of m pairs of points

$$\Omega_m = \Omega_m\left(c_{i_1}, ..., c_m\right) = \Omega_m\left(i_1, ..., i_m\right) =$$

$$\Omega_m\left(\left\{\mathbf{x}_{i_1}, y_{i'_1}\right\}, ..., \left\{\mathbf{x}_{i_n}, y_{i'_m}\right\}\right)$$

For instance, with triplets matching we get

$$\Omega_m = \Omega_m\left(c_i, c_j, c_k\right) = \Omega_m\left(\left\{\mathbf{x}_i, \mathbf{y}_{i'}\right\}, \left\{\mathbf{x}_j, \mathbf{y}_{j'}\right\}, \left\{\mathbf{x}_k, y_{k'}\right\}\right).$$

This is depicted in Fig. 2, and the affinity measure reflects the sort of matching we are aiming for. Local isometric matching, similar to the one used in [22], can be achieved by

$$\Omega_m\left(c_i, c_j, c_k\right) =$$

$$\exp - \frac{\left\|\overline{\mathbf{x}_i\,\mathbf{x}_j} - \overline{\mathbf{y}_{i'}\mathbf{y}_{j'}}\right\| + \left\|\overline{\mathbf{x}_i\,\mathbf{x}_k} - \overline{\mathbf{y}_{i'}\,\mathbf{y}_{k'}}\right\| + \left\|\overline{\mathbf{x}_k\,\mathbf{x}_j} - \overline{\mathbf{y}_{k'}\,\mathbf{y}_{j'}}\right\|}{\varepsilon^2}. \quad (3.1)$$

where $\overline{\mathbf{x}_i\,\mathbf{x}_j}$ is the length of the corresponding line segment. It was first shown by Zass and Shashua [40] that high order affinities can encode geometrical invariance, such as scale and affinity invariance (See Section 4.2). For instance, local scale invariance can be derived by computing the affinities Ω_3 of the triplets depicted in Fig. 2. Those are based on the difference in corresponding angles

$$\Omega_3\left(c_i, c_j, c_k\right) =$$

$$\exp - \frac{1}{\varepsilon^2}\left(\left\|\theta_1 - \theta_{1'}\right\| + \left\|\theta_2 - \theta_{2'}\right\| + \left\|\theta_3 - \theta_{3'}\right\|\right). \quad (3.2)$$

and are thus scale invariant.

The m-order affinities Ω_m, can be represented by a m-order tensor T_m, such that

$$T_m\left(i_1, ..., i_m\right) = \Omega_m\left(c_1, ..., c_m\right).$$

For instance, in triplets assignments, T_3 is given by a three-dimensional tensor

$$T_3\left(i, j, k\right) = \Omega_3\left(c_i, c_j, c_k\right). \quad (3.3)$$

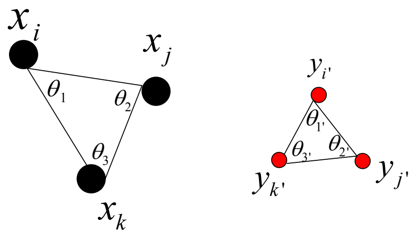


Figure 2: Scale invariant matching using high order affinities. The two triangles are differentiated by an apparent scale change. Matching using pairwise affinities will fail, while high order matching based on triplets will measure the discrepancy in the angles, and will thus be scale invariant.

The assignments are then recovered by computing the assignment vector $\mathbf{z}^*\in \{0,1\}^{n_1 K}$ that maximizes the overall assignment affinity

$$\mathbf{z}^* = \arg\max_{\mathbf{z}} \sum_{i_1,...,i_n} T_m\left(i_1,...,i_n\right)\mathbf{z}_{i_1}\mathbf{z}_{i_2}...\mathbf{z}_{i_n},\ \mathbf{z}^* \in \{0,1\}^{n_1 K}\,. \tag{3.4}$$

In the triplets case we get

$$\mathbf{z}^* = \arg\max_{\mathbf{z}} \sum_{i_1,...,i_n} T_3\left(i,j,k\right)\mathbf{z}_i\mathbf{z}_{\mathbf{j}}\mathbf{z}_k,\ \mathbf{z}^* \in \{0,1\}^{n_1 K}\,. \tag{3.5}$$

The tensor T_m is nonnegative and symmetric, such that

$$T_m\left(\mathbf{i}\right) = T_m\left(i_1,...,i_m\right) = T_m\left(\mathbf{i}'\right) \geq 0,$$

where $\mathbf{i}'$ is any permutation of $\mathbf{i}$. For a m-order tensor, we get $m!$ such permutations. For instance, for a three-way symmetric tensor $\mathbf{i} = (x,y,z)$ and

$$T_3\left(x,y,z\right) = T_3\left(x,z,y\right) = T_3\left(y,x,z\right) = \\ T_3\left(z,x,y\right) = T_3\left(z,y,x\right) = T_3\left(y,z,x\right).$$

This implies that instead of computing the $(n_1K)^m$ elements of T, it suffices to compute $\frac{(n_1K)^m}{m!}$ elements.

Solving Eqs. 3.4 and 3.5 is known to be np-complete, and analogue to the pairwise case [22], $\mathbf{z}^*$ can be estimated by computing the rank-one approximation (ROA) of T_m

$$T_m \approx \mathbf{w}\otimes_T\mathbf{w}...\mathbf{w}\otimes_{\mathbf{T}}\mathbf{w} \triangleq \prod_{i=1..m}\otimes_{\mathbf{T}}\mathbf{w}, \mathbf{w}\in\mathbb{R}^d, \tag{3.6}$$

where $\otimes_T$ is the *tensor* product. This will be rigorously justified in Section 3.2. The assignment $\mathbf{z}^*$ is derived by discretizing $\mathbf{w}$ the same way it was done in the pairwise case [22]. We discuss some alternatives in Section 3.4.

The effectiveness of this scheme is experimentally verified in Section 4. But still, a few issues have to be discussed. First, in the pairwise case the ROA is justified as the spectral relaxation of a np-complete problem [22]. Spectral relaxations of quadratic binary optimization problems are widely used and well understood, originating with the seminal work of Shi and Malik [37]. Unfortunately, this does not extend well to higher dimensions (tensors). The notions of ROA and eigendecomposition are different for tensors, from a theoretical a well as a numerical point of view.

To resolve these issues in the context of this work, we extend recent results by Zass and Shashua [40] on the probabilistic interpretation of affinity tensors. We show in Section 3.2 that the spectral decomposition is a proxy for the ROA of the affinity tensor T_m, and that it yields a maximum likelihood estimate of the HOA. The numerical aspects of tensor ROA are discussed in Section 3.1.

The second downside of the proposed HOA scheme is that practical assignment problem require extremely high memory storage and computational resources. For instance, in the image registration examples presented in Section 4 we match images using $n = 500$ points, each having $K = 4$ potential assignments. This results in an affinity tensor of dimensions $(n_1 K)^m = 2000^3 = 8 \cdot 10^9$ elements. Utilizing the supersymmetry it can be reduced to $\frac{(n_1 K)^m}{m!} = 1.3 \cdot 10^9$ elements. Computing and storing such volumes of data might render this scheme useless. Fortunately, the affinity tensor T_m is sparse, as significant nonzero elements would be those for which all assignments are true. Hence, only $\frac{1}{K^3} = 0.8\%$ of the affinity elements are expected to be nonzero. In practice, only 0.03% of the elements are nonzero. The increased sparsity is due to partial matching, as not all of the points $\mathbf{x}_i \in S_1$ can be matched to points in S_2, as some of them are due to clutter. The affinities of those clutter points will all be zero. To sparsify the affinity matrix numerically, we threshold it by $O\left(10^{-5}\right)$ as the affinities of true matches are typically of $O\left(10^{-1}\right)$. Yet, without having to deal with clutter we could have used significantly less points. It follows that the more points one needs to match due to clutter, the sparser the affinity tensor is.

The common approach for further sparsifying the affinity tensor in image matching problems, was proposed by Zass and Shashua [40] and Torresani *et al.* [38]. They propose to compute the high order affinities only for pixels of close proximity within the image. Thus, only a small subset of the affinities have to be computed. This heuristic is inapplicable to general HOA problems, not involving images. We present an alternative general random sparsification approach in Section 3.3.

In the following sections we discuss the tensor ROA and present several extensions to the basic scheme described above. These extensions make it computationally feasible in terms of storage and computational complexity, allowing us to run the image matching setup described above in less than two seconds.

3.1 Rank one approximation of nonnegative symmetric tensors

The study of tensors from an algorithmic as well applicative points of view has recently gained significant attention [19, 33]. By construction, the affinity tensor is nonnegative and symmetric, and we aim to compute its ROA. In the pairwise assignment case, we were able to apply the Perron-Frobenius theorem to the affinity matrix T_2. It guarantees the existence of a leading eigenvector and that all of its elements would be nonnegative. Using the Eckart-Young Theorem, it follows the leading eigenvector would also provide the optimal ROA. Numerically, the leading eigenvector can be computed by the Power Method that is robust and globally convergent.

Unfortunately, these properties do not carry over to tensors. A Perron-Frobenius like theorem for tensors was discussed by Lim [24] and the uniqueness of the lading eigenvector was proved by Chang *et al.* [7]. But, the leading eigenvector is not the optimal ROA [18]. Yet it provides a good approximation of it.

From a numerical perspective, there is a lack of robust ROA schemes, such as the matrix eigendecomposition schemes. A high order SVD algorithm was proposed by Lathauwer *et al.* [20] for asymmetric tensors, while a tensor ROA scheme, based on the Power method (HOPM) was suggested by the same authors in [21]. However, the HOPM is not convergent for symmetric tensors as claimed in [21] and demonstrated in [17]. A thorough survey of tensor decompositions was provided by Kolda and Bader [19].

We used a straightforward ROA scheme based on the HOPM [21]. We initialize the computation by unfolding the tensor into a matrix, as exemplified in Fig. 3 for an order-3 tensor. In that case, the leading left singular vector of the unfolded tensor $T_u \in \mathbb{R}^{(n_1 K)\times(n_1 K)^2}$ coincides with $\mathbf{x}$, which is the ROA of T_m. The right singular vector is given by $\begin{bmatrix}\mathbf{x}^T & ... & \mathbf{x}^T\end{bmatrix}^T \in \mathbb{R}^{(n_1 K)^2}$. In general, given a rank-one tensor T_m the unfolding results in a matrix $T_u \in \mathbb{R}^{(n_1 K)\times(n_1 K)^{m-1}}$ with a left singular vector $\mathbf{x}$. As the affinity tensor T_m is symmetric any axis can be chosen as the folding axis.

Due to the asymmetric dimensions of the matrix T_u, it is better to compute the leading eigenvector of $T_u T_u^T \in \mathbb{R}^{(n_1 K)\times(n_1 K)}$ that identifies with $\mathbf{x}$. Denote this initial estimate $\mathbf{x}_0$. Given $\mathbf{x}_0$, we apply 2-3 more HOPM iterations. This numerical scheme insures that the solution is nonnegative, as $\mathbf{x}_0$ will be nonnegative due to the Perron-Frobenius theorem applied to the matrix $T_u T_u^T$, and the HOPM consists of multiplications by the nonnegative elements of T_m.

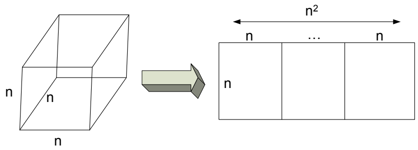


Figure 3: Tensor unfolding.

3.2 Probabilistic interpretation and Marginalized Tensor HOA

In this section we extend the probabilistic analysis of hypergraph matching by Zass and Shashua [40] to spectral matching. Zass and Shashua proposed a probabilistic model for pairwise matching, where the symmetric affinity tensor T_m is modeled by the *joint probability* of the assignments

$$T(\mathbf{i}) = P(\mathbf{i}), \mathbf{i} = [i_1, i_2 ... i_n]$$

Examining Eqs. 3.1 and 3.2, it follows that $0 \leqslant P(\mathbf{i}) \leqslant 1 \ \forall \mathbf{i}$ and that $P(\mathbf{i}) \approx 1$ implies that all the point correspondences are true. They further assume that the different assignments are statistically independent and show that the affinity tensor can be decomposed as

$$T_m = \prod_{i=1..m} \otimes_K \mathbf{p}, \mathbf{p} \in [0,1]^{n_1}, \tag{3.7}$$

where $\otimes_K$ is the Kronecker product.

Equation 3.7 constructs a matrix that can not be used for spectral matching. In particular, in the HOA problem applying Eq. 3.7 results in an affinity matrix rather than an affinity tensor T_m as in Eq. 3.6.

Hence we propose to apply tensor product to construct the tensor T_m

$$T_m = \prod_{i=1..m} \otimes_T \mathbf{p}, \mathbf{p} \in [0,1]^{n_1}, \tag{3.8}$$

where $\otimes_T$ is a tensor product. It follows that the true assignment probability is given by the ROA of T_m, without having to relate it to spectral relaxations, or discrete optimization. This justifies the HOA scheme based on the eigendecomposition of T_m which is equivalent to the one proposed by Duchenne *et al.* [11].

Following Section 3.1, we suggest to avoid computing the ROA of the tensor T_m due to its numerical inaccuracy. Instead we utilize the marginalization proposed by Zass and Shashua [40]. But instead of marginalizing T_m down to a vector as in their work, we propose to use *partial marginalization* by marginalizing T_m down to T_2 (a matrix) and computing its ROA by eigendecomposition. The eigendecomposition of matrices is optimal (Eckart-Young Theorem) and numerically stable [14].

Recalling that the affinity tensor is an approximation of the joint probability [40]

$$T_{m-1} = \sum_{i_1} T_m \left(i_1, i_2 ... i_{n_1}\right) =$$

$$\sum_{i_1} P\left(i_1, i_2 ... i_{n_1}\right) = P\left(i_2 ... i_{n_1}\right).$$

T_{m-1} is a symmetric tensor of order $m-1$ whose ROA identifies with the ROA of T_m. The marginalization is true for *any* joint probability function and does not assume statistical independence among the different assignments.

3.3 Randomly sparsified high order spectral matching

The affinity tensor is a sparse low rank tensor, and so is the unfolded matrix $T_u \in \mathbb{R}^{n \times n^{m-1}}$ discussed in Section 3.1, which is of rank one in the noise-free case. Hence, computing the leading left singular vector $\mathbf{x}$ is an overdetermined problem. For instance, Govindu [15] proposed to subsample (erase) some of the columns of T_u, while retaining $\mathbf{x}$ as the leading left singular vector. In image matching [40, 11] the particular structure of the problem can be utilized to prune the high order affinities a-priori, by only computing the affinities of close points.

In this section we propose a general approach to randomly sparsifying T_u and equivalently the original tensor T_m. This provides better accuracy than subsampling the columns of T_u [15] and can be applied to any high order matching problem. In particular, in image matching applications, it can be used alongside the spatial sparsification as in [40, 11].

There is a plethora of results on fast approximation algorithms for large low-rank matrices in general and singular value decompositions in particular [34, 23]. The work of Achlioptas and McSherry [1] is of particular interest as it provides the mathematical foundations for approximating the spectrum if a matrix given its sparsified replica. The core idea is that sparsifying a large low dimensional matrix is equivalent to adding a random matrix E whose entries are independent random variables with zero-mean and bounded variance

$$\widehat{T_u} = T_u + E.$$

The eigenvalues of such random matrices E are known to be of bounded magnitude [8] and do not influence the computation of the leading eigenvalues and eigenvectors of $\widehat{T_u}$. The more elements of T_u are erased, the larger the eigenvalues of the matrix E. Consider the eigendecomposition $E = \sum\limits_i \lambda_i \phi_i \phi_i^T$. The more nonzero elements in E, the larger its L_2 norm, and the larger will its eigenvalues λ_i be (recall that $\|\phi_i\| = 1$). Hence, this approximation is appropriate for sparse matrices T_u, whose further sparsification is equivalent to adding a noise matrix E with relatively small eigenvalues.

In the context of this work, T_u is sparsified by a factor Q during its construction, by drawing a random number $t \sim U\left[0,1\right]$ at each affinity entry and computing the affinity value only if $t < \frac{1}{Q}$. In order to retain the symmetry of T_u we apply the sparsification appropriately by making the sparsity pattern symmetric.

3.4 Soft assignment refinement and discretization

So far we suggested a high order matching scheme that results in a vector of soft assignments $\mathbf{w}^* \in \mathbb{R}^{n_1 K}$. Before discretizing $\mathbf{w}^*$ to yield the hard assignment $\mathbf{z}^*$, $\mathbf{w}^*$ can be refined by projecting it onto the space of soft assignment vectors. Cour *et al.* [10] utilize the spectral matching algorithm of Scott and Longuet-Higgins [35] as a spectral approach to inducing the matching constraints. First, $\mathbf{w}^*$ is reshaped as $W^* \in \mathbb{R}^{n_1 \times K}$ and the projection $\widehat{W}^*$ is given by

$$\widehat{W}^* = \arg\min_W \left\|W^* - W\right\|_2, \ s.t. \ W\underline{1} = \underline{1} \text{ and } W^T\underline{1} = \underline{1}. \tag{3.9}$$

Zass and Shashua [40] solve a similar problem as in Eq. 3.9, but the similarity of the assignment matrices is measured by the relative entropy measure.

Following Section 3.2, the soft assignment vector $\mathbf{w}^*$ (with or without refinement) identifies with the assignment probability P. We aim to compute the hard assignment $\mathbf{z}^* \in \{0,1\}^{n_1 K}$ that maximizes the overall assignment probability

$$\mathbf{z}^* = \arg\max_{\mathbf{z}} \left(\mathbf{z}^T \mathbf{w}^*\right) = \arg\max_{\mathbf{z}} \left(\mathbf{z}^T P\right) = \arg\max_{\mathbf{z}} \sum_{\mathbf{z}_k = 1} P\left(k\right), \mathbf{z}^* \in \{0,1\}^{n_1 K}$$
$$s.t.\ Z^* \underline{\mathbf{1}} \leqslant \underline{\mathbf{1}} \text{ and } \left(Z^*\right)^T \underline{\mathbf{1}} \leqslant \underline{\mathbf{1}}, \quad (3.10)$$

where $Z^* \in \{0,1\}^{n_1 \times K}$ is a row-wise reshaped replica of $\mathbf{z}^*$.

The greedy algorithm used by Leordeanu [22] provides an approximate solution of Eq. 3.10, by iteratively picking the highest element in $\mathbf{w}^*$ and inducing the assignment constraints. This can also be optimally solved by the Hungarian algorithm [31] .

4 Experimental results

We applied the proposed approach to synthetically generated random graphs as well as real images and sequences. In the synthetic graphs simulations we followed the experimental framework used by Zass and Shashua [40]. It allows to quantify the accuracy and resiliency of the proposed schemes, and compare it to contemporary state-of-the-art algorithms, by applying them to millions of randomly generated graphs. The same synthetic graph having a known ground truth, is used as input to all of the different matching schemes at a time. We also experiment with the registration of real images to show the applicability of the Tensor-HOA to image matching. Last, we analyze the matching of the *Hotel* and *House* sequences. We concentrate on using triplets affinities manifested by third-order tensors, as those are the most practical.

4.1 Matching of random synthetic graphs

In this set of trials we measure the accuracy and resiliency of our approach to different spatial deformations: noise, scale and outliers, both at the source and the target sets. In each trial we generated a set of

20 random points $S_1 = \{\mathbf{x}_i\}_{i=1}^{20}, \mathbf{x}_i \in \mathbb{R}^2, \mathbf{x}_i \sim U[0,1]^{2\times 1}$. We denote this randomly generated set as the *source set*.

In each trial a particular deformation (noise, scale or outliers insertion) is applied to S_1. We denote the distorted source set as the *target set* $S_2 = \{\mathbf{y}_i\}_{i=1}^{20}$, and apply different state-of-the-art matching schemes (including our own) to match S_1 to S_2. By construction, we have that $\mathbf{x}_i$ corresponds to $\mathbf{y}_i$ and this acts as the ground truth. We measure the matching accuracy as the number of correctly matched points divided by the total number of points that could potentially be matched. Note that in the outliers test, not every point $\mathbf{x}_i \in S_1$ has a corresponding point in S_2. In all simulations we used the triplets affinity measure given in Eq. 3.2 with the same kernel bandwidth $\varepsilon = \frac{\pi}{15}$. The affinity tensor was 10% full on average while thresholding the affinity tensor by 10^{-5}.

We tested various algorithmic setups that are detailed in Table 1, where each setup is made of a different combinations of the computational components. For a given graph distortion (i.e. scale, noise variance, number of outliers) we repeat the same experiment 100 times by generating 100 random sets S_1 (source graphs) and applying the distortion. We report the mean accuracy over the different trials.

Notation	Affinity	Margin.	Rank one approximation	T_1 Refinement
FA+TE	T_3 full affinity	none	Tensor RON + GradAssign	ProbOpt
FA+M2	T_3 full affinity	T_2 (matrix)	GradAssign	ProbOpt
FA+M1	T_3 full affinity	T_1 (vector)	none	ProbOpt
SA2+TE	T_3 sparse affinity 50%	none	Tensor ROA+GradAssign	ProbOpt
SA2+M2	T_3 sparse affinity 50%	T_2 (matrix)	GradAssign	ProbOpt
SA2+M1	T_3 sparse affinity 50%	T_1 (vector)	none	ProbOpt
SA5+TE	T_3 sparse affinity 20%	none	Tensor ROA+GradAssign	ProbOpt
SA5+M2	T_3 sparse affinity 20%	T_2 (matrix)	GradAssign	ProbOpt
SA5+M1	T_3 sparse affinity 20%	T_1 (vector)	none	ProbOpt
ProbOpt [40]	T_3 full affinity	T_1 (vector)	none	ProbOpt
Spectral [22]	T_2 full affinity	none	EIGS	none
GradAssign [13]	T_2 full affinity	none	GradAssign	none

Table 1: The different matching schemes. Greedy discretization was used as the last step in all schemes.

First we tested the robustness of our algorithm to additive random noise by computing the *target set* S_2 as

$$\mathbf{y}_i = \mathbf{x}_i + n_i, \forall \mathbf{x}_i \in S_1, n_i \sim N(0,\sigma).$$

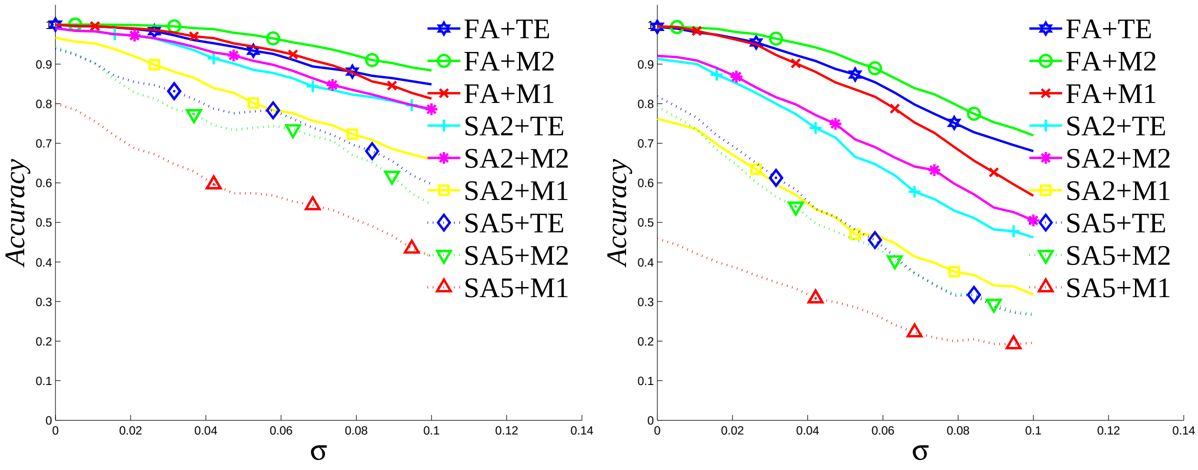
In applications such as image matching this simulates image distortions, as well as the localization inconsistencies of interest points detector. In this trial the matching is one-to-one. Figure 4 depicts the accuracy of the various schemes with respect to σ. In real matching scenarios descriptors can be used to prune the set of possible assignment per point $\mathbf{x}_i \in S_1$. Hence, we applied the different schemes with $K = 5$ and $K = 10$. Using $K = 5$ corresponds to having highly discriminative local features, able to significantly limit the number of potential assignments per point in S_1. The results depicted in Figs. 4a and 4b show that all of the schemes performed well, but for those using an aggressively sparsified affinity matrix (SA5+TE, SA5+M2,SA5+M1). In contrast, in image registration (Section 4.2), we sparsified the affinity tensor by a factor of 10 and were able to successfully register the images. This is due to a significant difference in the sparsity of the affinity tensors in both applications, and is further discussed in the next section.

The superior numerical stability of the matrix eigendecomposition compared to the tensor ROA is evident. Thus, the FA+M2 scheme outperforms the FA+TE, and SA2+M2 outperforms the SA2+TE scheme.

We compare the matching performance of the proposed approach to state-of-the-art schemes in Fig. 5. This is the same setup as in Fig. 4, but for sake of clarity we only show some of the Tensor-HOA formulations. The results of matching using $K = 10$ potential matches per $\mathbf{x}_i \in S_1$ are reported in Fig. 5a. It follows that the Tensor HOA schemes outperform all of the pairwise schemes, where the Graduated Assignment was the best. Figure 5b repeats this trial with $K = 20$. This implies that there is no pruning of potential matches and that the assignment problem is made more difficult. In this setup the Tensor HOA outperforms the Graduated Assignment and ProbOpt by 50%, while the pairwise spectral approach fails completely.

Scale invariant matching was our core motivation for developing the Tensor-HOA, as state-of-the-art pairwise matching schemes [22, 13] are unable to match sets of points related by scale changes. We drew 100 sets of points S_1, and induced a scale change ΔS and added noise

$$\mathbf{y}_i = \Delta S \cdot \mathbf{x}_i + n_i, \forall \mathbf{x}_i \in S_1, n_i \sim N(0, 0.05).$$



(a)

(b)

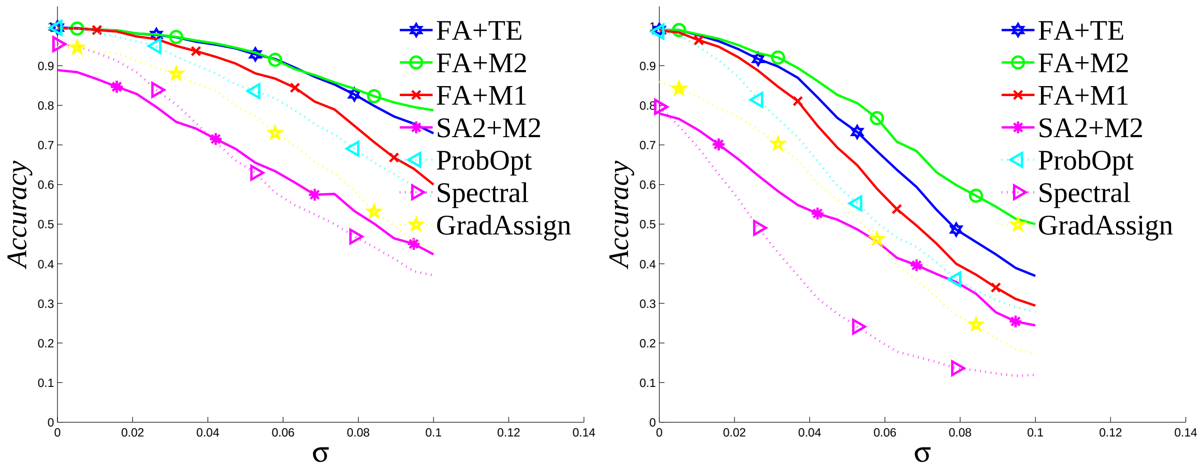
Figure 4: Graph matching simulations with varying additive noise. We added WGN to a set of 20 uniformly randomly spread points. The points were registered with (a) $K = 5$ (b) $K = 10$ potential matches per point.

The triplets affinity measure in Eq. 3.2 is scale invariant and can thus be applied. The results are shown in Fig. 6, where it follows that the tensor based schemes overcame the scale change. As in the noise resiliency trials, for $K = 5$ (Fig. 6a) all schemes performed similarly down to a few percents of accuracy. For $K = 10$ (Fig. 6b) we witness the same phenomenon as in the noise trials, where the aggressive sparsification (SA5) reduces the accuracy. The accuracy of the pairwise schemes reported in Fig. 7 quickly drops to that of random matching, where using $K = 10$ as (Fig. 7a) implies that random assignments will have an accuracy of 10%. The same applies to Fig. 7b.

Partial matching is of particular interest, as one-to-one matching is rarely present in real matching scenarios. Consider for instance the images used in Section 4.2. A feature points detector will detect points in one image, many of them lacking corresponding points in the other one. As before, we define an experimental set up that simulates the addition of outliers to both source and target sets.

We generate random sets with 20 points and add 10 random outliers to the source set S_1. Those have no correspondences in the second graph. Their potential K matches per point are picked randomly. For the inliers we insert the true match as one of the K potential matches. We also added a varying number of outliers to the target set, and random noise $N\left(0, 0.05\right)$.

The results are reported in Fig. 8. As before, the Tensor HOA schemes performed well for $K =$



(a)

(b)

Figure 5: Graph matching simulations with additive noise. We added WGN to a set of 20 uniformly randomly spread points. The points were registered with (a) $K = 10$ (b) $K = 20$ potential matches per point.

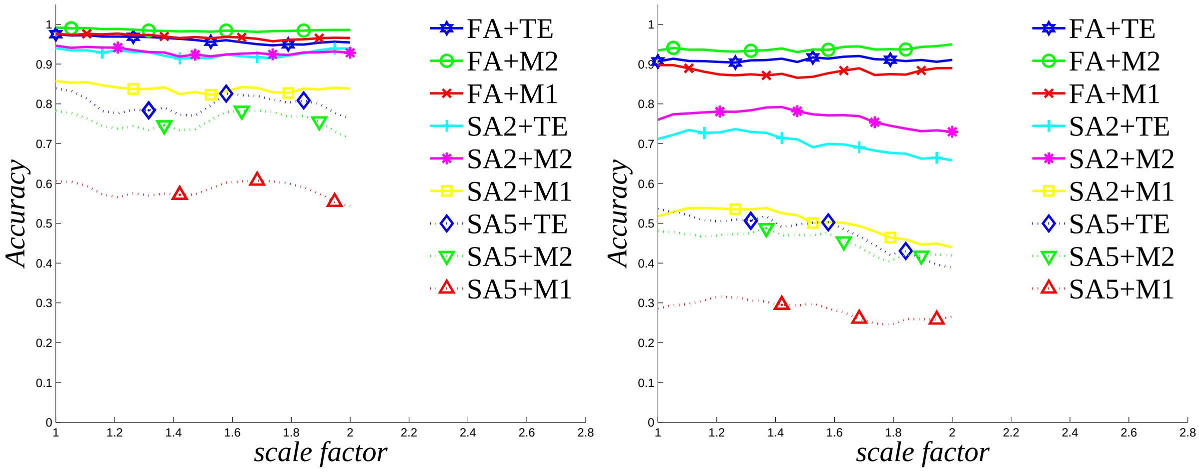
5. The results shown in Fig. 8a do not vary much among the different schemes. For $K = 10$ the sparsification hampers the accuracy again.

4.2 Image matching

The registration of images related by significant spatial deformation in general and scalings in particular, is an unresolved challenge in computer vision. This task requires to overcome scaling, noisy feature points localization and outliers present in the source as well as the target image. All image pairs used in this section are real images. In order to increase the scale differential we downscaled some of them.

In order to implement an image registration scheme by way of point assignment, we first have to represent the image as a set of interest points. Thus, we turn to image modeling by means of *local features* [26, 28]. These can be used to represent the image by a sparse set of salient points $\{x_i\}$ detected by a *detector*. Each interest point is represented by a vector of parameters denoted a *descriptor*. The descriptors are scale and rotation invariant. This is achieved by computing a *dominant scale* and *dominant angle* of each interest point x_i [26, 28]. Those allow to normalize the computation of the descriptors with respect to rotation and scale.

We detected 500 interest points in each image and computed their SIFT descriptors [26]. In each



(a)

(b)

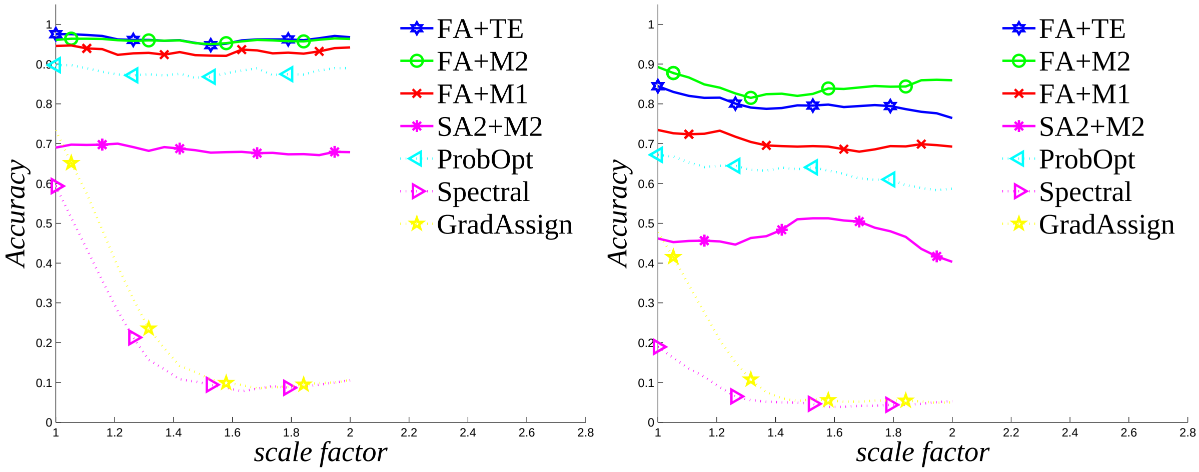
Figure 6: Graph matching simulations with varying scale and noise. A set of 20 uniformly randomly spread points was rescaled. The points were registered with (a) $K = 5$ (b) $K = 10$ potential matches per point.

figure we mark a subset of those on the images. One of the input images was chosen to contain the points $S_1 = \{\mathbf{x}_i\}_{i=1}^{500}$ while $S_2 = \{\mathbf{y}_i\}_{i=1}^{500}$ were taken in the other. In all registrations, we used a constant number of $K = 4$ potential correspondences for each point $\mathbf{x}_i$, chosen as the K points in S_2 with the most similar SIFT descriptors.

As image registration requires scale invariant matching, we used the scale invariant triplets affinity measure given in Eq. 3.2 with the same kernel bandwidth $\varepsilon = \frac{\pi}{15}$. Such an affinity measure can not be implemented with any pairwise scheme [22, 13] and is thus unique to HOA schemes.

The affinity tensor was thresholded by 10^{-5}. This yielded an affinity tensor that was on average $5 \cdot 10^{-4}$ full. To further sparsify it by a factor of 10, we applied the random sparsification scheme (Section 3.3), and got a tensor that was only $5 \cdot 10^{-5}$ full. In contrast, in the points matching trials in Section 4.1, the affinity tensor was 10^{-1} full on average. Hence, the sparsification was equivalent to adding a noise matrix E with $O\left(10^{-1} \cdot (n_1 K)^2\right)$ nonzero elements, compared to $O\left(10^{-4} \cdot (n_1 K)^2\right)$ such elements in this section. Following Section 3.3, this matrix E would have larger eigenvalues, and thus corrupt the ROA approximation.

The first example of image registration is given in Fig. 10 where the scale factor is approximately 2. It is typical for images with such scale changes to have a relatively small common support, which



(a)

(b)

Figure 7: Graph matching simulations with varying scale and noise. A set of 20 uniformly randomly spread points was rescaled. The points were registered with (a) $K = 10$ (b) $K = 20$ potential matches per point.

translates to a large number of interest points that are outliers.

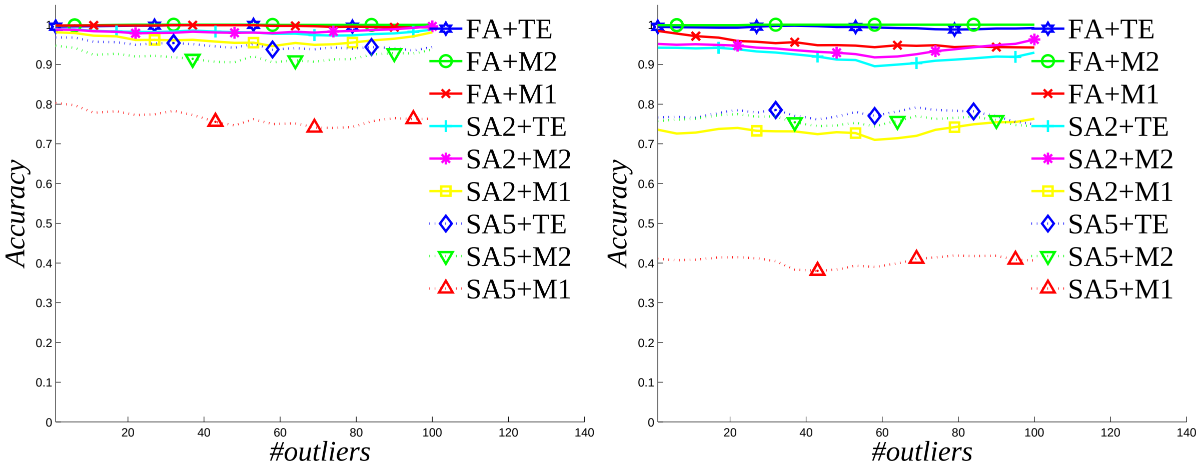
The registration of a pair of images related by a scaling factor of 2.8 and a change in the view angle is depicted in Fig. 11. The successful registration was able to overcome the significant variation in the viewing angle, that induced a projective transformation.

We further studied the scale invariance by increasing the scale differential of the image pair in Fig. 11. These results are showed in Fig. 12. We were able to successfully register the image in Fig. 12a having a scale factor of 6. But failed to estimate a scale factor of 8 in Fig. 12b. Since the common support and transformations between the image pairs in Figs. 12a and 12b are the same as those in Fig. 11, we attribute the failure to a break-down of the scale-invariance of the local descriptors.

The registration of images related by significant affine deformations was studied in Figs. 13 and 14. In Fig. 13 we successfully aligned images of planar objects under different viewing angles and a scale factor of 2.

In Fig. 14 we match Graffiti images taken from Oxford VGG database[1]. This is a set of six images where the affine distortion is increasing with respect to the first image of the series . We added an additional scaling of 2. The successful registration of the first and third images in the series is depicted

[1] http://www.robots.ox.ac.uk/∼vgg/data/data-aff.html



(a)

(b)

Figure 8: Graph matching simulations with added outliers. The points were registered with (a) $K = 5$ (b) $K = 10$ potential matches per point.

in Fig. 14a, while Fig. 14b shows the failure to register the first and fourth images. As in the registration failure in Fig. 12b, we attribute that to a reduction in the accuracy of the local descriptors due to the deformation. To further elucidate that we applied affine invariant SIFT features along side our Tensor HOA in Fig. 15. We successfully matched the first image in the Graffiti series to the fifth and sixth images, where as before, we downscaled the target image by a factor of 2. The relative deformation of those images is so severe that is was difficult to verify their accuracy visually.

To conclude, the proposed scheme was successfully applied to a variety of difficult image registration tasks. It seems that in practice, its accuracy is only limited by the resiliency and accuracy of the local features. Due the large number of outliers present in such images, the affinity tensor is sparse up to $O\left(10^{-4}\right)$ nonzero elements and can be further sparsified by applying random sampling.

4.3 Sequence matching

We applied our approach to the CMU *Hotel* and *House* sequences [2]. These were used in multiple works [11, 38] and provide a baseline for comparison. In order to evaluate the matching accuracy we tracked landmark feature points that were manually labeled and tracked in *all* frames. This allows to compare the performance of the different schemes over a varying temporal baseline - the larger the temporal

[2] http://vasc.ri.cmu.edu/idb/html/motion/hotel/index.html

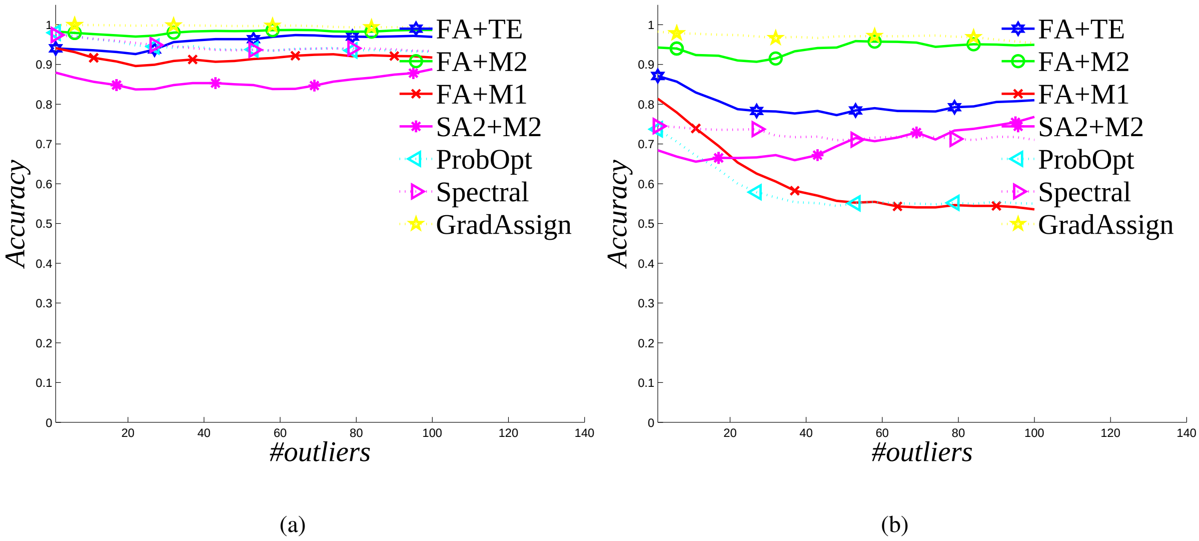


Figure 9: Graph matching simulations with added outliers. The points were registered with (a) $K = 10$ (b) $K = 20$ potential matches per point.

baseline (differential) between the frames, the larger the relative deformation, and the more difficult the matching. We matched each frame to the frames succeeding it and computed the average matching error per temporal baseline.

As in the noise tests in the previous section, we utilized the number of possible assignments K as a mean to vary the matching difficulty. For that we used shape-context shape descriptors [5] as a similarity measure between points. Hence, when matching a point in particular frame, it can only be matched to its K nearest shape descriptors.

The results for the *Hotel* sequence are depicted in Fig. 16. It consists of 101 frames and 30 landmark points, where its motion is close to rigid, and there are no significant scale changes or outliers. We vary $K = \{10, 20, 30\}$, where for $K = 10$ the matching problem is well constrained and all schemes perform similarly. The decline in the accuracy for large baseline values is due to the inaccuracy of the shape descriptors as the deformation became substantial. Hence, the true matching might not be within the $K = 10$ nearest neighbors of the SCs.

As K increases to $\{20, 30\}$ we witness the susceptibility of the pure spectral approaches (SM and FA+TE) to outliers, as their accuracy deteriorates. As expected, the proposed dual marginalization-spectral scheme (FA+M2) outperformed the FA+TE, due to the numerical instability of the eigendecomposition of tensors.

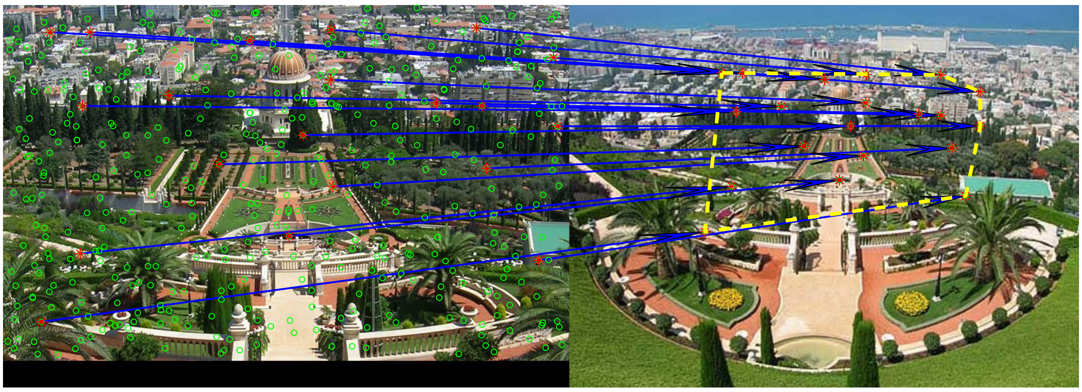


Figure 10: Fully automatic registration of images with a scaling factor of 2. Due to the significant scaling, the common image parts are small resulting in many interest points being outliers. The green circles are a subset of the local feature used.

We repeated the above experiment with the *House* sequences that consists of 30 landmark points and 111 frames. The motion in this sequence is more rigid than in the *Hotel* sequence and easier to match. This is confirmed by the results reported in Fig. 17. For $K = 10$ all algorithms were able to match all of the points, while for $K = 20$, the SM is the first to fail, while for $K = 30$ the GA also fails by up to 10%.

4.4 Implementation issues

The proposed scheme was implemented in Matlab, where the construction of the affinity tensor was coded in a MEX file, as it is loop-intensive. We used the Tensor Toolbox [4] to represent the sparse affinity tensors, and modified the HOPM solver within the Tensor Toolbox to efficiently handle symmetric tensors. For local features detectors and descriptors we used the open source Matlab implementation of Vedaldi and Fulkerson [39]. We also used the open source Matlab code of Timothee Cour [3] as a reference for implementing the spectral matching and the Graduated Assignment algorithms. For the Probabilistic Graph Matching of Zass and Shashua we used their publicly available Matlab code [4].

On average, the memory space required for matching 2 images was 40MB, needed for storing the sparse affinity tensor. It follows that without the use of the random sparsification this scheme would have

[3] http://www.seas.upenn.edu/~timothee/

[4] http://www.cs.huji.ac.il/~zass/gm/

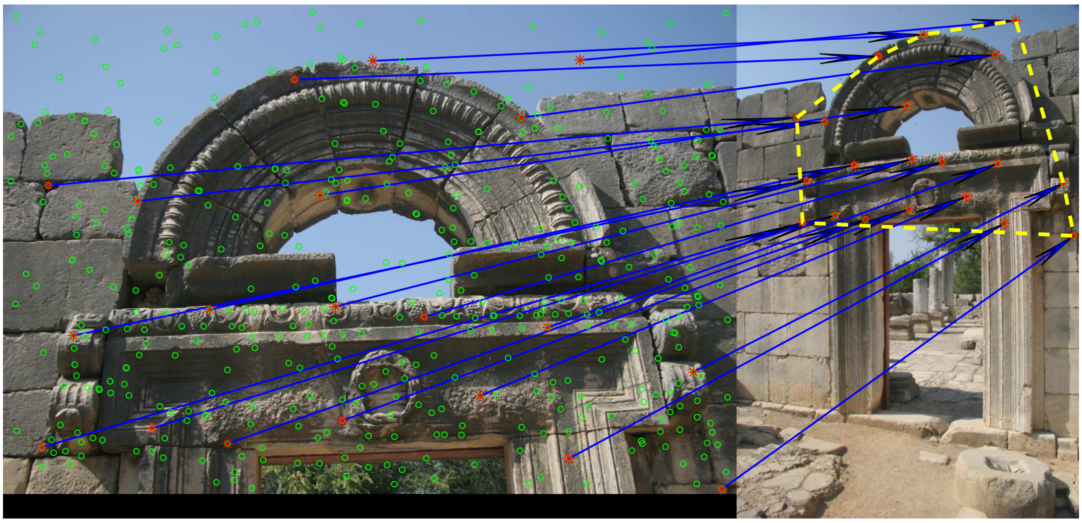
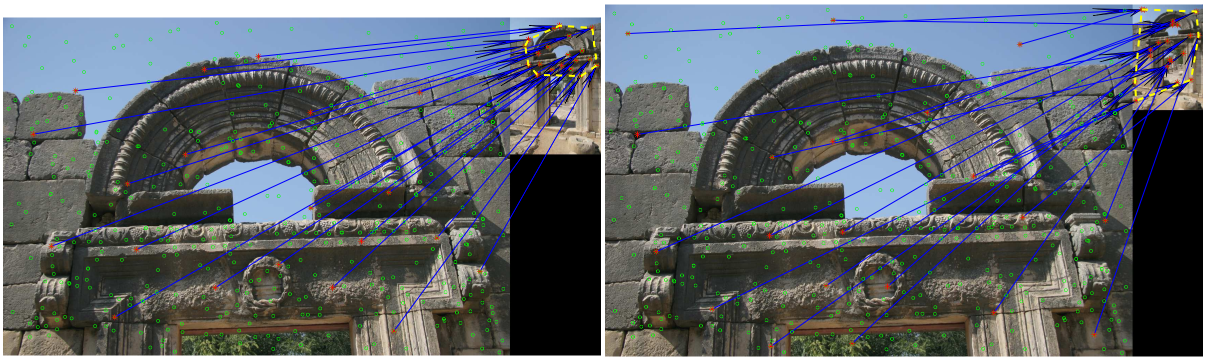


Figure 11: Regitration of images with a scale change of 2.8. Due to the different view angle and the projective transformation the scaling changes along the images. The green circles are a subset of the local feature used.

been significantly less practical. It typically took a second to construct the affinity tensor and another two seconds to compute its ROA. The timings were taken on a computer running an Intel Core2 Duo processor at 2.2GHz.

5 Summary and discussion

In this work we presented a general framework for solving high order assignment problems. The high order affinities are encoded in an affinity tensor Ω_m. It is based on the core assumption that the affinity measure Ω_m encodes the joint assignment probabilities. Ω_m is application specific and can be made invariant to certain deformations. We derived a marginalization scheme that maps a triplets tensor to matrix or a vector. By further assuming statistical independence among different assignments, we showed that the assignments can be recovered by the ROA of the affinity tensor/matrix. For tensors we used the HOPM to recover the ROA, as spectral relaxation does not extend well to tensors. We also showed how to sparsify the affinity tensor during its construction, making the Tensor-HOA computationally feasible. The scheme was shown to be effective in registering sets of points as well as real images.



(a) (b)

Figure 12: Registration of image with significant scale changes. We are able to recover a scale change of 6 in (a), but fail to estimate a scale of 8 in (b). Both use the same image as in Fig. 11. The green circles are a subset of the local feature used.

References

[1] D. Achlioptas and F. Mcsherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54(2):9, 2007.

[2] S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *International Conference On Machine Learning (ICML)*, pages 17–24, Pittsburgh, PA, 2006.

[3] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *CVPR*, volume 2, pages 838–845, San Diego, CA, 2005.

[4] B. W. Bader and T. G. Kolda. Matlab tensor toolbox version 2.2, January 2007.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[6] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 26–33, Washington, DC, USA, June 2005. IEEE Computer Society.

[7] K. C. Chang, K. Pearson, and T. Zhang. Perron-frobenius theorem for nonnegative tensors. *Communications in Mathematical Sciences*, 6(2):507–520, December 2008.

[8] F. Chung. *Spectral graph theory*. Number 92. CBMS-AMS, May 1997.

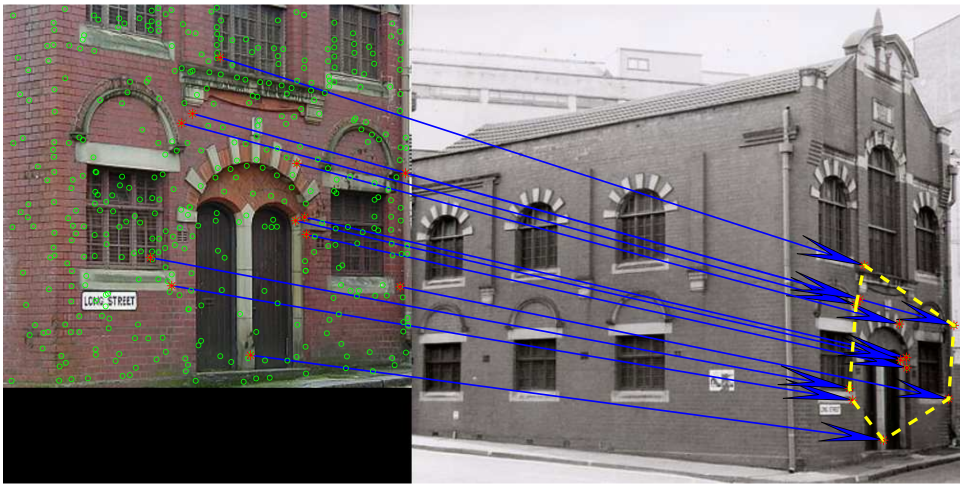


Figure 13: Registration of images related by a scale factor of 2 and a significant projective distortion due to the difference in the viewing angle. The green circles are a subset of the local feature used.

[9] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(3):265–298, 2004.

[10] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 313–320. MIT Press, Cambridge, MA, 2007.

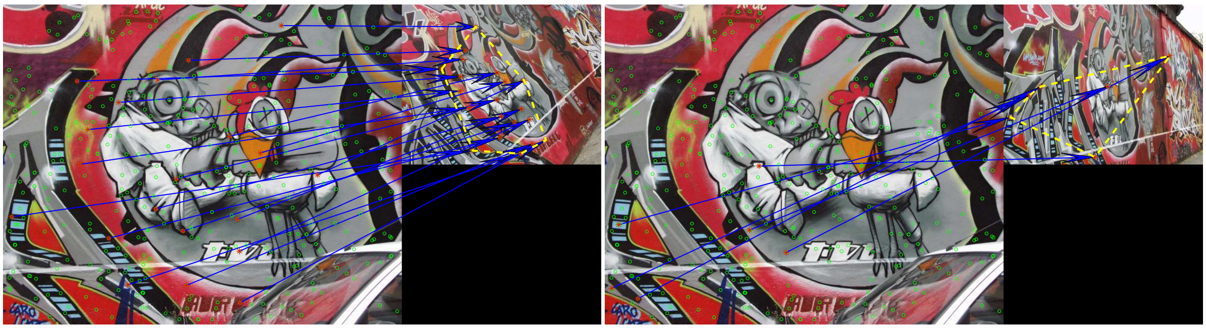
[11] O. Duchenne, F. Bach, I. Kweon, and J. J. Ponce. A tensor-based algorithm for high-order graph matching. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1980 – 1987, June 2009.

[12] A. Egozi, Y. Keller, and H. Guterman. A probabilistic approach to spectral graph matching. *Submitted*, 2009.

[13] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.

[14] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[15] V. M. Govindu. A tensor decomposition for geometric grouping and segmentation. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1150–1157, 2005.



(a)

(b)

Figure 14: Matching images from the Graffiti series. We downscaled the target image by a factor of two. In (a) we successfully matched the first image to the forth one. In (b) we failed to match the first image to the fifth one that has a more significant distortion. We used regular SIFT descriptors to match the images. The green circles are a subset of the local feature used.

[16] J. H. Hays, M. Leordeanu, A. A. Efros, and Y. Liu. Discovering texture regularity via higher-order matching. In *9th European Conference on Computer Vision*, pages 522–535, May 2006.

[17] E. Kofidis and P. A. Regalia. On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM J. Matrix Anal. Appl.*, 23(3):863–884, 2001.

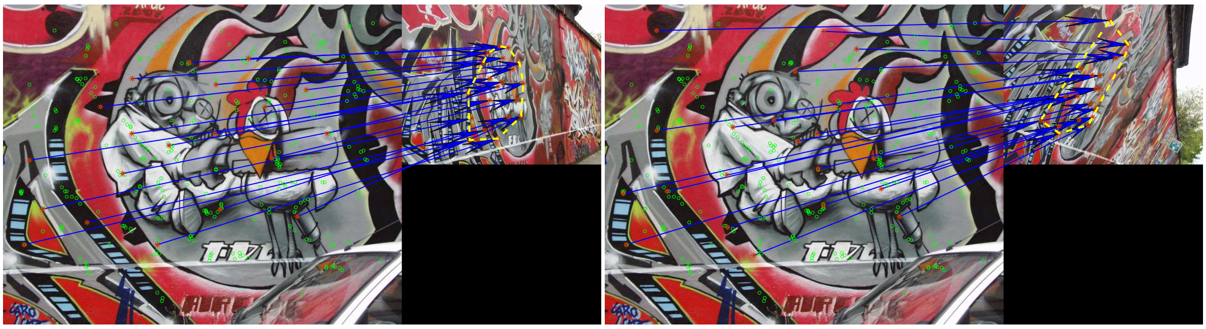
[18] T. G. Kolda. A counterexample to the possibility of an extension of the Eckart-Young low-rank approximation theorem for the orthogonal rank tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 24(3):762–767, January 2003.

[19] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3), September 2009 (to appear).

[20] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[21] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. On the best rank-1 and rank-(r1,r2,. . .,rn) approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342, 2000.

[22] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *International Conference of Computer Vision (ICCV)*, volume 2, pages 1482 – 1489, October 2005.



(a) (b)

Figure 15: Image registration using the Tensor-HOA and affine invariant image features. In (a) we successfully registered the first image in the Graffiti series to the fifth and in (b) we registered the first to the sixth. The green circles are a subset of the local feature used.

[23] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, pages 0709640104+, December 2007.

[24] L.-H. Lim. Singular values and eigenvalues of tensors: A variational approach. In *Proceedings of the 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing.*, pages 129–132, December 2005.

[25] C. K. Liu, A. Hertzmann, and Z. Popović. Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics*, 24(3):1071–1081, August 2005.

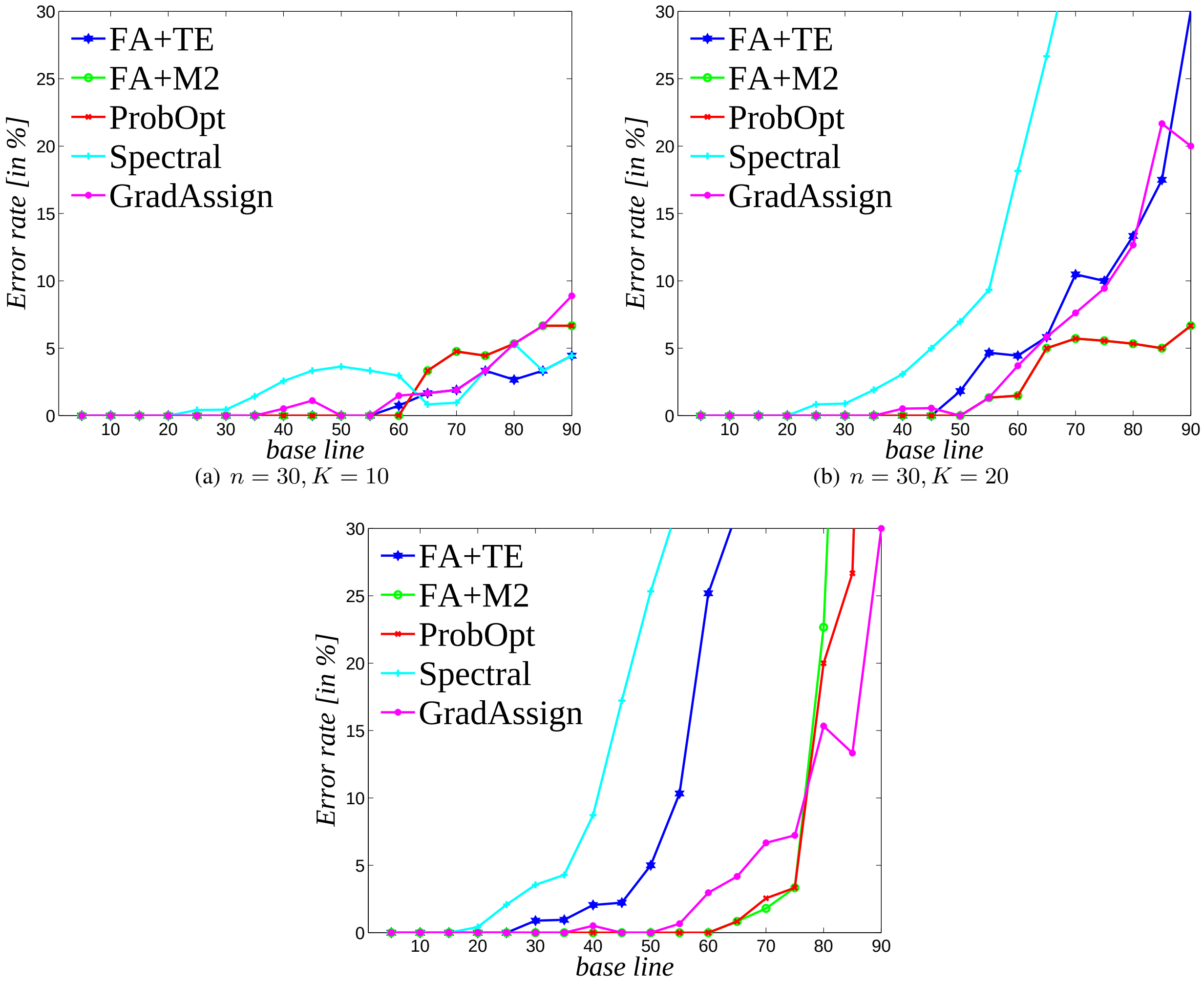
[26] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[27] a. Maciel, Jo and a. P. Costeira, Jo˙ A global solution to sparse correspondence problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):187–199, 2003.

[28] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, October 2005.

[29] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[30] H. Mirzaalian, G. Hamarneh, and T. Lee. A graph-based approach to skin mole matching incorporating template-normalized coordinates. In *Computer Vision and Pattern Recognition, 2009. CVPR*



(a) $n = 30, K = 10$

(b) $n = 30, K = 20$

(c) $n = 30, K = 30$

Figure 16: *Hotel* series analysis results. We tracked 30 points using a varying number of K possible assignments.

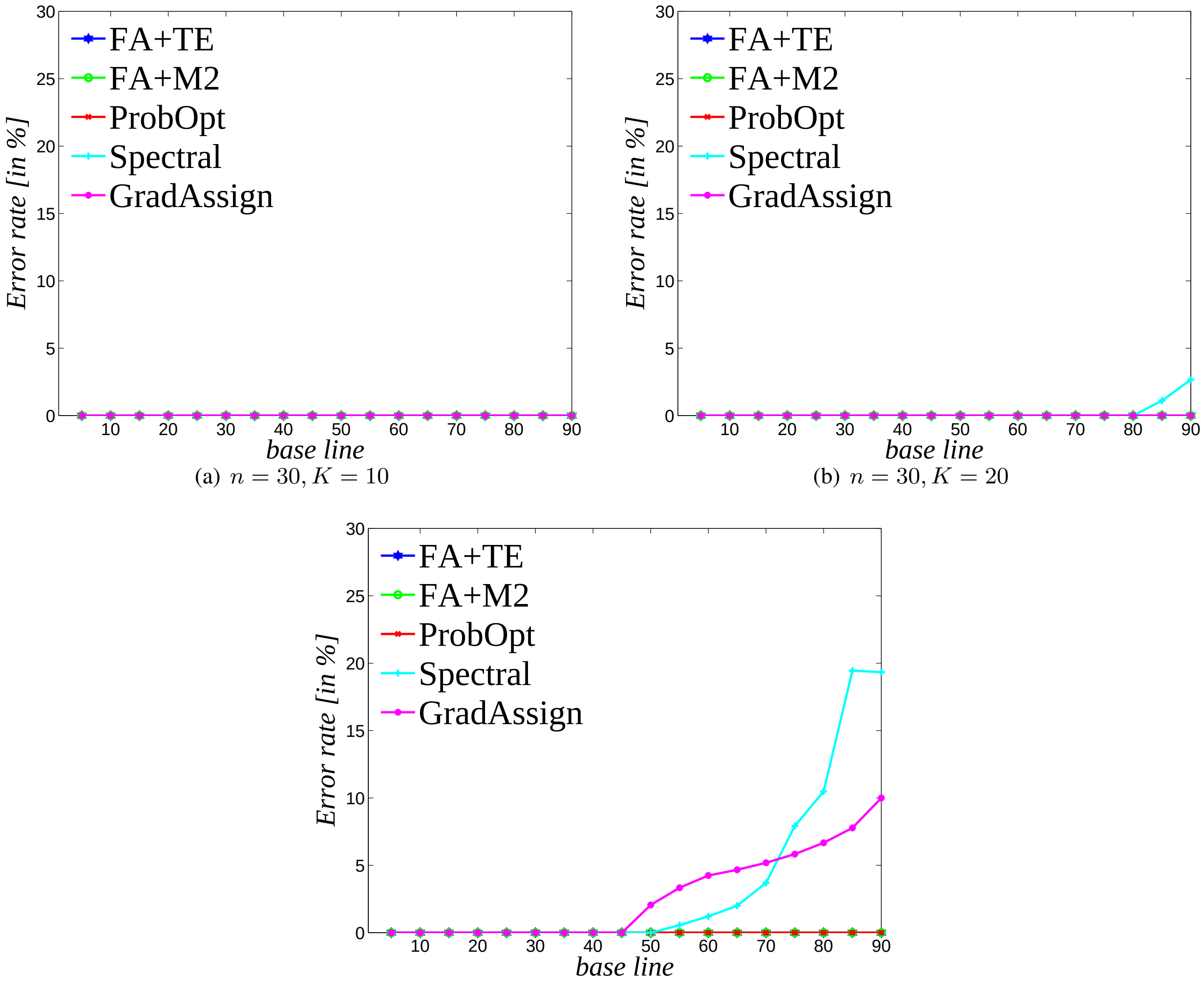
2009. IEEE Conference on, pages 2152 – 2159, June 2009.

[31] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[32] J. Philbin and A. Zisserman. The oxford buildings dataset. http://www.robots.ox.ac.uk/ vgg/data/oxbuildings/index.html.

[33] C. Pierre, G. Golub, L.-H. Lim, and B. Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.*, 30(3):1254–1279, 2008.

[34] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages



(a) $n = 30, K = 10$

(b) $n = 30, K = 20$

(c) $n = 30, K = 30$

Figure 17: *House* series analysis results. We tracked 30 points using a varying number of K possible assignments.

143–152, Washington, DC, USA, 2006. IEEE Computer Society.

[35] G. Scott and H. Longuet Higgins. An algorithm for associating the features of two images. In *Royal Society London*, volume B-244, pages 21–26, 1991.

[36] A. Shashua, R. Zass, and T. Hazan. Multi-way clustering using super-symmetric non-negative tensor factorization. In A. Leonardis, H. Bischof, and A. Pinz, editors, *ECCV (4)*, volume 3954 of *Lecture Notes in Computer Science*, pages 595–608. Springer, 2006.

[37] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[38] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models

and global optimization. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 596–609, Berlin, Heidelberg, 2008. Springer-Verlag.

[39] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[40] 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, June 2008.

[41] D. Zhou, J. Huang, and B. Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1601–1608. MIT Press, Cambridge, MA, 2007.