

# Unsupervised Visual Alignment with Similarity Graphs

Fatemeh Shokrollahi Yancheshmeh, Ke Chen, and Joni-Kristian Kämäräinen  
Department of Signal Processing, Tampere University of Technology, Finland  
{fatemeh.shokrollahiyancheshmeh, ke.chen, joni.kamarainen}@tut.fi

## Abstract

Alignment of semantically meaningful visual patterns, such as object classes, is an important pre-processing step for a number of applications such as object detection and image categorization. Considering the expensive manpower spent on the annotation for supervised alignment methods, unsupervised alignment techniques are more favorable especially for large-scale problems. Fine adjustment can be effectively and efficiently achieved with image congealing methods, but they require moderately good initialization which is largely invalid in practice. Alignment of visual class examples with large view point changes remains as an open problem. Feature-based methods can solve the problem to some degree, but require manual selection of a good seed image and omit the fact that examples of a semantic class can be visually very different (e.g., Harley-Davidsons and Scooters in “motorbikes”). In this work, we overcome the aforementioned drawbacks by defining visual similarity under the generalized assignment problem which is solved by fast approximation and non-linear optimization. From pair-wise image similarities we construct an image graph which is used to step-wise align, “morph”, an image to another by graph traveling. We automatically find a suitable seed by novel centrality measure which identifies “similarity hubs” in the graph. The proposed approach in the unsupervised manner outperforms the state-of-the-art methods with classes from the popular benchmark datasets.

## 1. Introduction

Visual alignment of an image ensemble is to find the corresponding control points between them. This is an important pre-processing step for a number of high-level computer vision applications such as object detection and categorization [1, 11, 15, 23, 37]. In those applications, it can be made difficult due to the large pose variation of class examples in the images. The alignment remains hot yet challenging topic especially in large-scale visual recognition problems to i) avoid the manpower spent on annotating the control points or object landmarks in supervised object align-

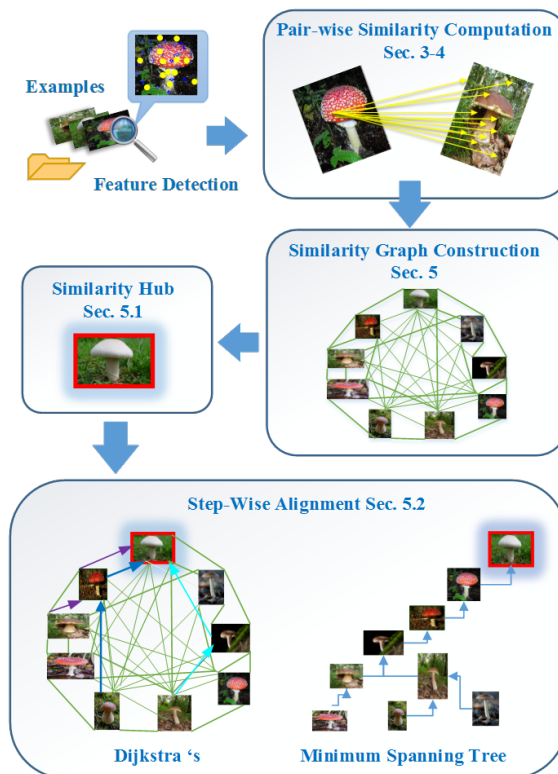


Figure 1. The workflow of our visual alignment approach.

ment [46], ii) the lack of moderately good initial alignments needed for image congealing algorithms [16], and iii) the manual seed selection in the feature-based alignment [26].

The recent “big visual data” databases, such as ImageNet [38] and COCO [27], have made it possible to train effective class detectors with minimal annotation, only class labels, and avoid over-fitting despite the huge number of model parameters [24]. It is still unclear whether this is due to a few quantized poses of classes, but certain undesirable properties indicate that the problem is not completely solved [40]. Prior to the large datasets and deep neural networks, the part-based methods requiring additional annotation of bounding boxes achieved state-of-the-art [10, 11, 25]. Bounding boxes have their own drawbacks which the best part-based methods partly overcome

by hacks, such as bounding box clustering [11], to identify discrete sub-classes or poses (*e.g.*, vertical/horizontal guitar). The hacks work well with limited poses, but fail with objects of similar spatial dimensions and suffer from unbalanced data between sub-classes. It is clear that even stronger annotation, such as explicit object poses [45] or landmarks [1, 23] improve detection. The downside is that more extensive manual annotation is needed.

An alternative solution to manual annotation is unsupervised visual alignment of object class images. Accurate alignment can be effectively and efficiently achieved with the recent image congealing methods [29, 16, 43, 9, 17]. These, however, require moderately good initialisation which is largely invalid in practice. An alternative approach is the recent feature-based alignment [26], but nevertheless, the main shortcoming of the feature-based approach is the need to manually select and search a good seed image, which is still dependent on the efforts dedicated by humans. Moreover, the both methods may fail in the case of visual sub-classes under the same semantic label, such as scooters, sport bikes and Harley-Davidsons in “motorbikes”.

In this work, we adopt the feature-based approach, but to overcome the aforementioned drawbacks devise pair-wise visual similarity as an assignment problem which is solved by fast approximation and non-linear optimization. From pair-wise similarities we construct an image graph which is used to step-wise align, “morph”, an image to another by graph traveling (see Figure 1). Our method also automatically finds a suitable seed by novel centrality measure which identifies “similarity hubs” in the graph. The proposed approach in the unsupervised manner outperforms the state-of-the-art methods with classes from the popular benchmark datasets. Source codes and data for our experiments will be made publicly available<sup>1</sup>.

## 2. Related Work

Our interest is on semantic level visual alignment and thus we omit works related to alignment of different view points of the same scene (stitching) [4] and methods for specific classes, such as faces [7, 34, 19]. Special cases are also cross-domain matching [39], non-rigid registration [5] and temporary alignment [21].

**Graph-Based Representation** – Graph-based representation has been employed in various works of visual object matching [20, 44, 36, 30] utilizing graph structures to represent a set of images. However, a graph is usually used only to represent spatial configuration of features between two images [20], or connections omit the spatial constellation (Bag-of-Words) [44] or the purpose is to match specific objects classes [36, 30]. To the authors’ best knowledge, our

work is the first to model pairwise similarity as the generalized assignment problem and represent “group similarity” in a full-connected similarity graph.

**Image Congealing** – The semantic level alignment gained momentum after the seminal work of Learned-Miller [29]. Its extensions [16, 43, 9, 17] provide effective and efficient fine alignment after moderately good initial alignment. Congealing methods stack images to perform gradual transformations to optimize stack similarity. Replacing pixels with local features (*e.g.*, SIFT [28]) provides better robustness to imaging distortions [16, 17]. An approach not requiring initial alignment is based on spatial verification of local features [35] and was proposed in [26]. The problem in this feature-based congealing is the manual selection of a good seed which may require testing all images and the slow RANSAC based spatial verification. Our approach uses features, but defines the similarity under a generalized assignment framework. The seed selection is solved by our novel centrality measure and the similarity graph.

## 3. Feature-Based Similarity

In this and the next section we use the term “cost” despite the fact that the term used in the titles is “similarity”. This discrepancy is done purposely since many related works build upon matching costs and, for example, Euclidean distance provides an intuitive measure how well features or their coordinates match. However, after building the optimization framework we switch to similarity in Section 4.2 where we propose our fast approximation algorithm for feature-based image similarity.

Our feature-based similarity (cost) is motivated by the part-based representations successfully adopted in visual class detection [12, 25, 11]. The quality of feature-based alignment of two images  $I_a$  and  $I_b$  is measured in two ways: how similar feature points are to their corresponding feature points, and how much the spatial arrangement of the feature points is changed. The matching function can be divided to the feature match and feature geometric distortion costs

$$C(I_a, I_b) = \lambda_1 C_{match}(I_a, I_b) + \lambda_2 C_{dist.}(I_a, I_b) , \quad (1)$$

where  $\lambda_1$  and  $\lambda_2$  are the trade-off parameters between the two terms. Equivalents of (1) have been used in object matching [3], searching sketches from images [2] and non-rigid matching of image sequences [41]. The work [2] and [41] solve the problem with the assumption of small geometric distortions and [3] requires manually segmented exemplars of stored classes.

Our feature-based image representation consists of  $N$  feature descriptors  $F_{i=1\dots N}$  (*e.g.*, SIFT [28]) and their spatial coordinates  $x_{i=1\dots N}$ . The cost of matching two images

<sup>1</sup>[https://bitbucket.org/kamarain/imgalign\\_code/](https://bitbucket.org/kamarain/imgalign_code/)

in (1) can be written as

$$C(I_a, I_b) = \lambda_1 C_{match}(F^{(a)}, F^{(b)}) + \lambda_2 C_{dist.}(\mathbf{x}^{(a)}, \mathbf{x}^{(b)}) . \quad (2)$$

The form in (2) separates feature matching and geometric distortion, but is misleading since the two are dependent. The cost function implicitly assumes the matching variables known: the assignments  $\mathbf{A}_{N_a \times N_b}$  and the geometric transformation  $\mathbf{T}$ . The assignment matrix elements  $a_{ij}$  define which feature  $F_i^{(a)}$  of  $I_a$  correspond to which feature  $F_j^{(b)}$  of  $I_b$ . The transformation  $\mathbf{T}$ , such as a  $3 \times 3$  linear homography matrix, transforms points from the space of  $I_b$  to the space of  $I_a$ .  $\mathbf{A}$  provides evidence of feature visual appearance match and  $\mathbf{T}$  of geometric distortion, therefore a more accurate definition of the cost is

$$\begin{aligned} C(I_a, I_b) &:= C(I_a, I_b; \mathbf{T}, \mathbf{A}) = \\ &C\left(\left\{F^{(a)}, \mathbf{x}^{(a)}\right\}, \left\{F^{(b)}, \mathbf{x}^{(b)}\right\}; \mathbf{T}, \mathbf{A}\right) = \\ &\lambda_1 C_{match}\left(F^{(a)}, \mathbf{A}F^{(b)}\right) + \lambda_2 C_{dist.}\left(X^{(a)}, \mathbf{A}T(X^{(b)})\right) \end{aligned} \quad (3)$$

where the feature matching cost term depends of the feature descriptors and the assignment  $\mathbf{A}$ , and the geometric distortion cost term depends on the assignment and the transformation  $\mathbf{T}(\cdot)$ . The assignment can be achieved by matrix multiplication and a practical example of the transformation is a  $T_{3 \times 3}$  homography matrix where the transformation  $\mathbf{T}(\cdot)$  includes the mappings between the non-homogeneous and homogeneous coordinates.

## 4. Similarity Algorithm

### 4.1. Problem Formulation

The similarity cost in (3) can be used to find the pairwise similarity value of the images  $I_a$  and  $I_b$  for a given geometric transformation  $\mathbf{T}$  and assignments in  $\mathbf{A}$ . The practical problem, however, is

$$C(I_a, I_b; \mathbf{T}, \mathbf{A}) = \min_{\mathbf{T}, \mathbf{A}} C(I_a, I_b) .$$

By defining the transformation  $\mathbf{T}$  as a problem parameter we can write the minimization problem as

$$\begin{aligned} &\text{minimize } \sum_i \sum_j c_{ij} a_{ij} \\ &\text{subject to } \sum_j a_{ij} \leq 1 \quad i = 1, \dots, N_a \\ &\quad \quad \quad \sum_i a_{ij} \leq 1 \quad j = 1, \dots, N_b \\ &\quad \quad \quad a_{ij} \in \{0, 1\} \end{aligned} \quad (4)$$

where the assignment costs  $c_{ij} = C(i, j)$  can be computed given the transformation  $\mathbf{T}$  and the descriptors  $F^{(a)}$

and  $F^{(b)}$ . The most straightforward solution is to adopt the  $N_a \times N_b$  descriptor and location distances

$$D^F(i, j) = \|F_i^a - F_j^b\|, \quad D^X(i, j) = \|\mathbf{x}_i^a - \mathbf{T}(\mathbf{x}_j^b)\| \quad (5)$$

and combine them as

$$C = \lambda_1 D^F + \lambda_2 D^X .$$

The form in (4) has the global optimum at the trivial solution  $a_{ij} = 0 \quad \forall i, j$ . To avoid the trivial solution one needs to enforce all features (of  $I_a$ ) to be mapped to some feature which is achieved by changing the inequality to equality:

$$\Rightarrow \sum_j a_{ij} = 1 \quad i = 1, \dots, N_a . \quad (6)$$

To allow features without correspondences, outliers, one also needs to introduce  $N_a$  ‘‘dummy assignments’’  $F^\epsilon$  that represent the outliers, *i.e.* the corresponding feature of  $F_a$  is not assigned to any of the features in  $F_b$  but to the dummy outlier with a fixed cost  $\epsilon$ :

$$\begin{aligned} F_j^b &= F^\epsilon \text{ and } c_{ij} = \epsilon \\ &\text{for } i = 1, \dots, N_a, j = N_b + 1, \dots, N_b + N_a \end{aligned} \quad (7)$$

With the above extensions (4) reduces to *the assignment problem* for which  $O(N^3)$  solvers such as the Hungarian method exist [33].

As an alternative solution, we can avoid the dummy assignments and estimation of their costs in (6) and (7) by changing the minimization of the similarity cost  $C$  to the maximization of the similarity  $S$  and rewrite (4) as

$$\begin{aligned} &\text{maximize } \sum_i \sum_j s_{ij} a_{ij} \\ &\text{subject to } \sum_j a_{ij} \leq 1 \quad i = 1, \dots, N_a \\ &\quad \quad \quad \sum_i a_{ij} \leq 1 \quad j = 1, \dots, N_b \\ &\quad \quad \quad a_{ij} \in \{0, 1\} \end{aligned} \quad (8)$$

The form in (8) avoids the dummy variables and yields to the feature assignment that provides the largest similarity value (under the transformation  $\mathbf{T}$ ). The maximization problem is known as *the generalized assignment problem*, which is NP-hard and even APX-hard to approximate [6]. Next we introduce our fast approximation of the maximization problem with the computational complexity of  $O(N)$ .

### 4.2. Approximation

We can map the distances (costs) (5) in  $[0, \infty]$  to similarity values in  $[0, 1]$  by the exponential function

$$S(i, j) = e^{C(i, j)} = e^{\lambda_1 D^F(i, j)} e^{\lambda_2 D^X(i, j)} = S^F(i, j) S^X(i, j) .$$

---

**Algorithm 1** Generalized assignment approx. solution
 

---

- 1: Compute the feature distance matrix  $\mathbf{D}_{N_a \times N_b}^F$  (e.g., SIFT[28]).
  - 2: On each row of  $\mathbf{D}^F$  set the  $K$  smallest to 1 and 0 otherwise.
  - 3:  $\mathbf{S}^X = \mathbf{0}$ .
  - 4: **for**  $i = 1 : N_a$  (features of  $I_a$ ) **do**
  - 5:   Compute the distance from  $\mathbf{x}_i^{(a)}$  to  $\mathbf{T}(\mathbf{x}_j^{(b)})$  for  $j = 1, \dots, K$  non-zero entries of  $\mathbf{D}^F$  and if  $\mathbf{D}^X(i, j) \leq \tau_X$  then set  $\mathbf{S}^X(i, j) = 1$  and break.
  - 6: **end for**
  - 7: **return** the number of non-zero terms in  $\mathbf{S}^X$
- 

In the exponential form  $\lambda_1$  and  $\lambda_2$  define the similarity decay from the exact match.  $\lambda_2$  can be defined in the spatial domain with intuitive interpretation, but defining  $\lambda_1$  in the feature space is difficult as, for example, structure of the SIFT feature space is unclear.

From the computational point of view it is wasteful to compute similarity values in the cases where either  $\mathbf{S}^F$  or  $\mathbf{S}^X$  or both are close to zero. To sparsify the similarity matrix we replace the exponential function with the Heaviside step function  $\mathcal{H}(\cdot)$  (i.e. unit step function, a discontinuous function whose value is zero for negative argument and one for positive argument) as

$$S(i, j) = \mathcal{H}(\mathbf{D}^F(i, j) - \tau_F) \mathcal{H}(\mathbf{D}^X(i, j) - \tau_X) \quad (9)$$

where  $\tau_F$  and  $\tau_X$  are Heaviside thresholds which define the points where similarities go from 1 to 0. The form in (9) provides substantial speedup for the two reason:

1. the first term does not depend on the transformation  $\mathbf{T}$  but is constant and can be computed in advance and
2. only the  $\mathbf{S}^X$  entries for which the  $\mathbf{S}^F$  entries are non-zero need to be computed.

To avoid measurements in the complex feature space we replace the feature distances in  $\mathbf{S}^F$  with their *rank-order distances*, i.e.  $\mathbf{S}^F$  entries are 1 for the  $\tau_F = K$  best feature matches in  $I_b$ . With this setting the similarity matrix  $S(i, j)$  is sparse binary of  $KN_a$  non-zero entries. To speedup the computation even further we may cascade the computation with a fixed spatial threshold  $\tau_X$  and stop after the first point below the threshold has been found. With these settings the minimum number of computations needed is  $N_a$  and the maximum  $KN_a$  that sets the final computational complexity of our approximate assignment solution to  $O(N)$ . In the experiments, the values  $K = 5$  and  $\tau_X = 0.02$  were found good, where  $\tau_X$  is made resolution independent by dividing the distance with the image diagonal. The full solver can be written with a few lines of pseudo-code shown in Algorithm 1.

By setting the type of the transformation  $\mathbf{T}$  to 2D similarity, we have the four degrees of freedom: *translation* ( $x, y$ ), *rotation*  $\phi$ , and *scaling* ( $s$ ). This is only a four dimensional search space in which we can efficiently utilize the well-

known Nelder-Mead nonlinear optimization technique [31]. The combination of the Nelder-Mead optimization on top of the fast approximation of the generalized assignment problem provides fast computation of the pair-wise image similarity  $S(I_a, I_b)$ .

## 5. Similarity Graph

Using the previously defined method to compute pair-wise image similarities of  $N$  images we can construct a full  $N \times N$  image similarity matrix

$$\mathbf{G}(i, j) = S(I_i, I_j)$$

that is a *weighted adjacency matrix* of a full connected graph  $G$ . A graph describing the visual similarities between examples from the Caltech-101 motorbikes class are shown in Figure 2. Similarities computed using the approxima-

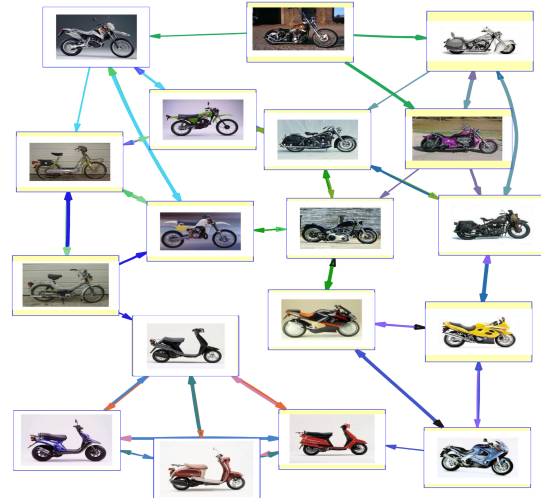


Figure 2. A motorbikes similarity graph constructed from the pair-wise similarities  $S(I_i, I_j)$  (strongest 10% links plotted).

tion algorithm are not symmetric since the rank-order based maximization of  $S(I_i, I_j)$  is bounded by the feature cardinality of  $I_i$ . At this stage we do not exploit asymmetry properties but enforce the weighted adjacency matrix symmetric:

$$\mathbf{G}(i, j) = \max(\mathbf{G}(i, j), \mathbf{G}(j, i)) \quad .$$

For the symmetric  $\mathbf{G}$  most graph algorithms, such as the *minimum spanning tree (MST)* become available, but again we found the rank-order statistics based representation more effective in our experiments:

$$\mathbf{G}(i, j) = \frac{N}{\text{rank}(\mathbf{G}(i, j), \text{sort\_ascend}(\mathbf{G}(:, j)))} \quad .$$

Value 1 denotes strong (short graph distance) and value  $N$  low similarity (long graph distance). With the above similarity value transformation the weighted adjacency matrix  $G$  represents a full-connected undirected graph.

### 5.1. Similarity “Hubs”

By the definition of *structural centrality of graphs* the graph  $G$  contains nodes, images, that often appear on the shortest paths between two random images  $I_i$  and  $I_j$  [13, 14]. By automatically identifying these “alignment hubs” we can select good candidates to which other images are accurately aligned. Such hubs correspond to manually selected “seeds” in [26].

Inspired by the random walk closeness centrality [32] we define a centrality measure based on the first and second order similarity statistics of each image node:

$$\mu_i = \frac{1}{N} \sum_j G(i, j), \quad \sigma_i = \frac{1}{N-1} \sqrt{\sum_j (G(i, j) - \mu_i)^2} .$$

To identify nodes with exceptional similarity to other nodes, the single node statistics are compared to the average statistics over all nodes

$$\mu = \frac{1}{N} \sum_i \mu_i, \quad \sigma = \frac{1}{N-1} \sqrt{\sum_j (\mu_i - \mu)^2} .$$

We select a set of central hubs  $H$  using the statistical test of one-sided normal distribution: ( $1 \times stdev$  corresponding to the 16% best values):

$$H = \{I_i\} \text{ for which } \mu_i \geq \mu + \sigma .$$

To select a single image  $I'$  from  $H$  to which the rest are aligned we switch to the second order statistics by taking the node with the smallest similarity variance

$$I' = \operatorname{argmin}_{I_i \in H} \sigma_i .$$

### 5.2. Step-Wise Alignment to the Central Hubs

All images can be aligned to a single space which is the most central hub image  $I'$  identified by the centrality computation proposed in Sec. 5.1. The possible alignment strategies exploiting the graph  $G$  structure are the following:

- Direct alignment [26],
- The minimum spanning tree (MST) path (e.g. the Prim’s algorithm [8]), or
- The shortest graph path (e.g. the Dijkstra’s algorithm [8]) .

In our experiments the first option of the direct alignment provides poor results especially in the cases where input images contain examples from visually different sub-classes.

In these cases the two other strategies that utilize graph traveling are more effective and allow “morphing” between visually dissimilar images, e.g., from a “scooter” to “Harley-Davidson”. The results are demonstrated in the experiments.

## 6. Experiments

In our experiments we used the same datasets (r-Caltech-101 and Labeled Faces in the Wild) and performance measures from the most recent alignment and congealing works [26, 17] for which the authors provide code to run their methods with the aforementioned benchmarks. In addition, we selected and annotated landmarks to challenging classes from the ImageNet dataset. In principle, evaluations are based on manually annotated landmarks which ideally map to the same locations and the alignment error is zero. However, for more than 3 landmarks exact mapping is not anymore guaranteed and in that case our “ideal result” represents the best possible alignment by the annotated landmarks themselves. In all experiments our descriptors are detected using the dense SIFT in the VLFeat toolbox [42].

### 6.1. Comparison with the State-Of-The-Arts

In our first experiment, the benchmark in [26] was run using the provided data and landmarks. The benchmark is particularly suitable for the feature-based alignment method [26] and unsuitable for the congealing method by Huang *et al.* [17] (randomized Caltech-101 has been introduced by the same authors [22]). In Figure 3 are the results for the ideal alignment (with manual landmarks), the two state-of-the-art methods, and our method. The x-axis is the average mean squared error of all landmarks after the alignment and the y-axis represents the number of images (max. 50) for which the specific alignment accuracy was achieved. In general,  $\leq 0.05$  is excellent,  $\leq 0.10$  is good and  $\leq 0.15$  is satisfactory in the terms of the image diagonal normalized distance measure used in the face detection literature. It is noteworthy, that for all cases our unsupervised method is superior to the feature-based method where the optimal seed image was manually selected. The congealing fails completely as the images are artificially misaligned with large translation, rotation and scale changes.

### 6.2. Computational Burden

In this experiment we replaced the RANSAC spatial matching algorithm of Lankinen *et al.* [26] with our fast assignment algorithm and the Nelder-Mead optimization (Sec. 4.2) and compared the accuracy and computing times. The results for the same four classes as in the previous experiment are collected to Table 1. Note that in the table we have fixed the operation point (x-axis) to 0.10 representing good alignment accuracy and report the proportion of images for which the accuracy was achieved. Our method is

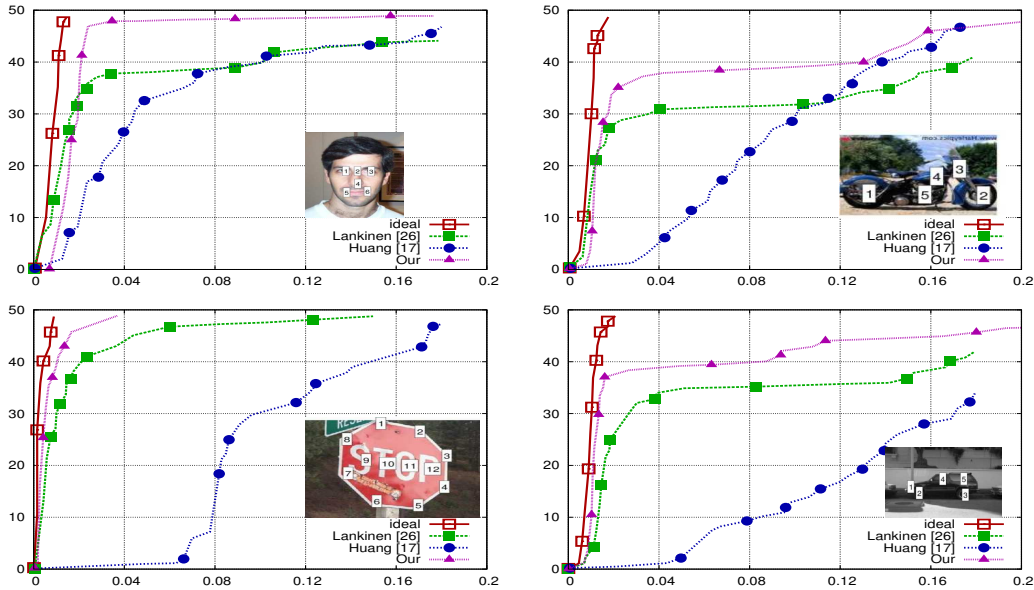






Figure 3. Our method vs. state-of-the-art for the benchmark in [26] (r-Caltech-101 Faces, motorbikes, stop sign and cars).

Table 1. The proportions of correctly aligned images (accuracy threshold 0.10) and the computation times for the feature-based alignment (FB) in [26] with the original RANSAC matching and with our fast assignment solver in Sec. 4.2.

				
FB [26] acc.	86%	76%	98%	74%
comp. time (s)	170	61	163	96
FB with Alg. 1	86%	76%	100%	80%
comp. time (s)	83	27	60	45

consistently 2-3 times faster and produces the same or better accuracy. The results validate that our theoretical framework for similarity optimization is more effective and efficient than the heuristic RANSAC matching in [26].

### 6.3. Stepwise Alignment

For evaluation of the two stepwise strategies, the Dijkstra shortest path algorithm and Prim’s minimum spanning tree (MST) algorithm (Sec. 5.2), we compared them for Caltech-101 and ImageNet classes. Interestingly, the results were often rather complementary; images misaligned with Dijkstra were correctly aligned with MST and vice versa. In addition, it was found that with minor geometric variance Dijkstra was better (faces in Figure 4), but with significant geometric variation MST produced better alignment (starfish). In general, Dijkstra is preferred due to more reliable results.

In addition to the stepwise strategies, the direct alignment can be used as well. That, however, completely failed in the presence of many sub-classes, is very sensitive to the

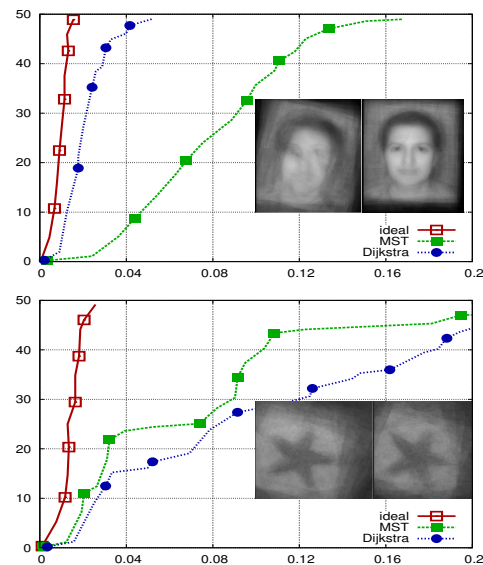


Figure 4. Comparison of the MST and Dijkstra algorithms for the stepwise alignment of Caltech faces and starfish images (left: average image of MST, right: Dijkstra).

hub selection and even in good conditions was on average inferior to the MST and Dijkstra stepwise alignments. That is demonstrated in Figure 5 for the two ImageNet classes. Note that Meerkats contain 3D pose changes which make also the ideal result clearly worse as compared to the previous examples from the Caltech datasets.

### 6.4. Face Verification

In addition to the alignment experiments we run the face verification experiments from [17] with the Labeled Faces

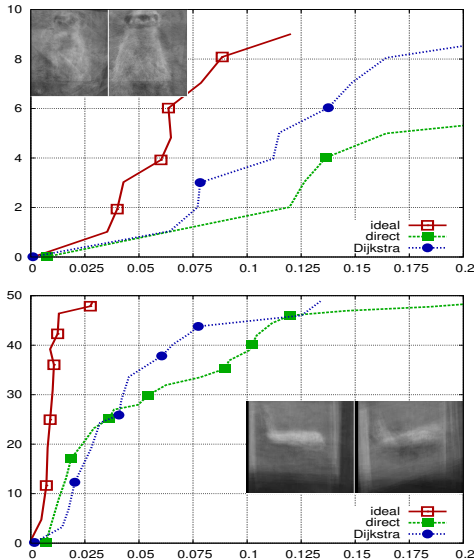


Figure 5. Direct vs. stepwise (Dijkstra) alignment of ImageNet classes airplanes and meerkat (left: direct, right: stepwise).

Table 2. The Labeled Faces in the Wild face verification benchmark [17] with aligned images.

Alignment	Avg. Accuracy
Original images	57.1%
Congeaing [17]	64.2%
Direct (Ours)	53.6%
Dijkstra’s (Ours)	71.4%

in the Wild database (LFW) [18]. Since the large number of face images makes the dataset very “dense” over geometric transformations, we made the benchmark more difficult by randomly sampling a 150 identities sub-set at time, performing training and testing on the sub-set, and averaged the results. The face bounding box of the automatically identified hub image is transformed to other face images using the best hub features and after that we run the matching algorithm from [17] using their code (details can be found from the original article). The results for unaligned, congealed and our stepwise aligned images are in Table 2 where our method with Dijkstra step-wise alignment is the best.

## 7. Conclusions

In this work, we have investigated the problem of unsupervised image alignment and, in particular, aligning images of object class examples. The previous approaches of congealing and feature-based matching suffered from manual seed selection, lack of good initial alignment and visually distant sub-classes. We defined a pairwise image similarity measure that combines both local part similarity and geometric distortion (Sec. 3) and the actual similarity value can be

found by searching the maximum of the similarity function. The search was cast as a combination of the generalized assignment problem and non-linear optimization (Sec. 4.1) for which we proposed an effective and efficient approximation in Sec. 4.2. To solve the sub-class and seed selection problems we constructed a full-connected similarity graph where the seed was identified as a “similarity hub” (Sec. 5.1) where all image can be “morphed” using alignment jumps over the graph nodes (Sec. 5.2). In the experiments, our method outperformed the state-of-the-art semi-supervised (manual seed selection) feature-based method and state-of-the-art unsupervised image congealing.

## Acknowledgments

This work was funded by Academy of Finland under the Grant No. 267581, D2I SHOK project funded by Digile Oy and Nokia Technologies (Tampere, Finland) and the Doctoral School PhD grant provided by Tampere University of Technology. The authors wish to acknowledge CSC - IT Center for Science, Finland for generous computational resources.

## References

- [1] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*. Springer, 2012. 1, 2
- [2] S. Bagon, O. Brostovski, M. Galun, and M. Irani. Detecting and sketching the common. In *CVPR*, 2010. 2
- [3] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005. 2
- [4] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int J Comput Vis*, 74(1), 2007. 2
- [5] F. Brunet, V. Gay-Bellile, A. Bartoli, N. Navab, and R. Malgouyres. Feature-driven direct non-rigid image registration. *Int J Comput Vis*, 93, 2011. 2
- [6] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Information Processing Letters*, 100(4), 2006. 3
- [7] T. F. Cootes, C. J. Twining, V. S. Petrovi, K. O. Babalola, and C. J. Taylor. Computing accurate correspondences across groups of images. *PAMI*, 32(11), 2010. 2
- [8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009. 5
- [9] M. Cox, S. Sridharan, S. Lucey, and J. Cohn. Least squares congealing for large number of images. In *CVPR*, 2009. 2
- [10] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 1
- [11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9), 2010. 1, 2

- [12] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Int J Comput Vis*, 61(1), 2005. [2](#)
- [13] L. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1), 1977. [5](#)
- [14] L. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1, 1979. [5](#)
- [15] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013. [1](#)
- [16] G. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, 2007. [1](#), [2](#)
- [17] G. Huang, M. Mattar, H. Lee, and E. Learned-Miller. Learning to align from scratch. In *NIPS*, 2012. [2](#), [5](#), [6](#), [7](#)
- [18] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007. [7](#)
- [19] I. Kemelmacher-Shlizerman and S. Seitz. Collection flow. In *CVPR*, 2012. [2](#)
- [20] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. In *CVPR*, 2008. [2](#)
- [21] G. Kim and E. Xing. Jointly aligning and segmenting multiple web photo streams for the inference of collective photo storylines. In *CVPR*, 2013. [2](#)
- [22] T. Kinnunen, J.-K. Kamarainen, L. Lensu, J. Lankinen, and H. Kälviäinen. Making visual object categorization more challenging: Randomized Caltech-101 data set. In *ICPR*, 2010. [5](#)
- [23] I. Kokkinos and A. Yuille. Unsupervised learning of object deformation models. In *ICCV*, 2007. [1](#), [2](#)
- [24] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. [1](#)
- [25] M. Kumar, A. Zisserman, and P. Torr. Efficient discriminative learning of parts-based models. In *ICCV*, 2009. [1](#), [2](#)
- [26] J. Lankinen and J.-K. Kamarainen. Local feature based unsupervised alignment of object class images. In *BMVC*, 2011. [1](#), [2](#), [5](#), [6](#)
- [27] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. [1](#)
- [28] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision. The proceedings of the seventh IEEE international conference on*, volume 2, 1999. [2](#), [4](#)
- [29] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities of transforms. In *CVPR*, 2000. [2](#)
- [30] H. Myeong, J. Y. Chang, and K. M. Lee. Learning object relationships via graph-based context model. In *CVPR*, 2013. [2](#)
- [31] J. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7, 1965. [4](#)
- [32] J. Noh and H. Rieger. Random walks on complex networks. *Phys. Rev. Lett.*, 92(118701), 2004. [5](#)
- [33] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Dover Publications, 2nd edition, 1998. [3](#)
- [34] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images. In *CVPR*, 2010. [2](#)
- [35] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. [2](#)
- [36] J. Philbin, J. Sivic, and A. Zisserman. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *Int J Comput Vis*, 95(2), 2011. [2](#)
- [37] E. Riabchenko, J.-K. Kämäräinen, and K. Chen. Learning generative models of object parts from a few positive examples. In *ICPR*, 2014. [1](#)
- [38] O. Russakovsky, J. Deng, Z. Huang, A. C. Berg, and L. Fei-Fei. Detecting avocados to zucchinis: what have we done, and where are we going? In *ICCV*, 2013. [1](#)
- [39] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. In *SIGGRAPH Asia*, 2011. [2](#)
- [40] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. [1](#)
- [41] M. Toriki and A. Elgammal. One-shot multi-set non-rigid feature-spatial matching. In *CVPR*, 2010. [2](#)
- [42] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008. [5](#)
- [43] A. Vedaldi and S. Soatto. A complexity-distortion approach to joint pattern alignment. In *NIPS*, 2006. [2](#)
- [44] S. Xia and E. R. Hancock. Incrementally discovering object classes using similarity propagation and graph clustering. In *ACCV*, 2009. [2](#)
- [45] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. [2](#)
- [46] X. Xiong and F. De la Torre Frade. Supervised descent method and its applications to face alignment. In *CVPR*, 2013. [1](#)