

# Contour Correspondence via Ant Colony Optimization

Oliver van Kaick\*, Ghassan Hamarneh†, Hao Zhang‡, Paul Wighton§  
School of Computing Science, Simon Fraser University, Burnaby, BC Canada

## Abstract

We formulate contour correspondence as a *Quadratic Assignment Problem (QAP)*, incorporating proximity information. By maintaining the neighborhood relation between points this way, we show that better matching results are obtained in practice. We propose the first *Ant Colony Optimization (ACO)* algorithm specifically aimed at solving the *QAP-based shape correspondence problem*. Our *ACO framework* is flexible in the sense that it can handle general point correspondence, but also allows extensions, such as order preservation, for the more specialized contour matching problem. Various experiments are presented which demonstrate that this approach yields high-quality correspondence results and is computationally efficient when compared to other methods.

## 1. Introduction

Finding a meaningful matching between shapes is a fundamental problem in geometry processing, with many applications in computer graphics, vision, and medical imaging. In this paper, we focus on 2D contour correspondence, a classical problem in computer vision for object tracking, recognition, and retrieval [1, 11, 24, 25], among other tasks. In medical computing, establishing point correspondence allows for statistical shape modelling and analysis of anatomical structures [7]. Contour matching is also the first step towards planar shape morphing [16], which finds applications in animation and shape analysis. Even in 3D shape modeling, the matching of contours is often an integral subproblem. For instance, surface reconstruction from CT or MRI data, or from data collected via the lofting technique, requires correspondence between contours from adjacent slices [19]. Also, reducing the 3D object matching problem to the matching of a set of projected object outlines [5] was shown to be effective for 3D shape retrieval [29].

One of the main approaches to contour correspondence is to compute a shape descriptor for each selected feature point to be matched. A matching can then be extracted from the descriptors in a variety of ways, e.g., via the simple greedy best matching [12], solved as a bipartite matching problem using the Hungarian method [21], relying on variants of the iterative closest point (ICP) scheme [3], which is based on descriptor distances and shape alignment via rigid or non-rigid deformations [1, 6], or computed by dynamic programming under point ordering [16, 24].

However, a drawback of all these optimization schemes, along with most other contour correspondence algorithms proposed so far, is that they treat the shape descriptors independently and do not consider *proximity* information measured between feature points on the same shape. For example, such information may be used to ensure that a feature pair which is close-by on one shape gets mapped to points that are also close on the other shape. Compared to approaches based purely on shape descriptors, the use of proximity information, e.g. by incorporating a regularization term in the cost function, can provide a better handling of shapes with missing parts or a lack of salient features.

Incorporating proximity into an optimization framework, we can formulate point correspondence via the *Quadratic Assignment Problem (QAP)* [22]. It is well known that QAP is one of the most difficult optimization problems to solve, yet a simple heuristic which mimics the behavior of ants has shown a great deal of success [31]. In this paper, we adopt *ant colony optimization (ACO)* [8], which we review in Section 3, to solve the contour correspondence problem and make the following contributions.

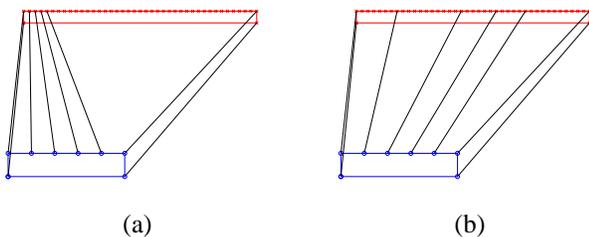
- We formulate the general point correspondence problem in terms of QAP (Section 4.1), incorporating proximity information, and propose the *first ACO algorithm* to compute the matching (Section 4.2). Note that this basic ACO framework is applicable to the matching of unorganized 2D point sets.
- Specific to contour matching, we extend our basic framework to enforce *order preservation* (Section 4.3).
- We show that the consideration of proximity has a different effect on contour matching compared to merely

\*GrUVi Lab. Email: ovankaic@cs.sfu.ca.

†Medical Image Analysis Lab. Email: hamarneh@cs.sfu.ca.

‡GrUVi Lab. Email: haoz@cs.sfu.ca.

§Medical Computing Lab. Email: pwighton@cs.sfu.ca.



**Figure 1. Example of a matching computed preserving order, in (a), compared to one preserving proximity and order, in (b).**

enforcing order preservation, as done previously via dynamic programming [16, 24]. An advantage of the former is shown in Figure 1; further demonstration of the feasibility and effectiveness of our ACO approach is given through numerous experiments (Section 5).

Note that while order preservation can be strictly enforced for contour correspondence [16, 24], none of the proposed solutions are applicable to the matching of surfaces due to a lack of canonical ordering among points residing on a 2-manifold. However, our basic QAP and ACO framework can still be extended to the surface setting by replacing proximities measured along a contour to proximities measured over a surface, e.g., using geodesic distance. We shall leave this topic for future studies but will offer some discussions on its potential in Section 6.

## 2. Related work on correspondence

While the problem of 3D shape matching and retrieval is receiving much attention lately [33], research on 2D shape matching [34] has remained active due to the problem’s inherent difficulty and wide-ranging applications. Motivated by problems from image analysis, many algorithms are designed to match unorganized sparse point sets, e.g., [1, 3, 6, 17]. Although these point matching techniques can often be directly applied to the contour correspondence problem, contours define explicit ordering between the points, which may be taken advantage of to improve performance [16, 24, 25]. In this section, we briefly survey existing contour matching paradigms and other point correspondence schemes that are applicable to contour matching. An overview of ACO, our adopted framework for contour correspondence, is given in the next section.

Many contour matching algorithms rely on local shape descriptors: a descriptor at a point  $p$  along a contour typically encodes the position or other geometric properties at the point itself (e.g., local curvature) or properties of the

remaining contour points, from the perspective of  $p$ . A recent survey and performance evaluation of local descriptors is given by Mikolajczyk and Schmid [20]. The focus of our work is on establishing correspondence regardless of the local descriptor used. However, in this paper, we mainly use shape context [1] as the local shape descriptor. Shape context records a histogram of uniformly placed point samples along a contour, parameterized by distance and angular extent with respect to the point described.

With a shape descriptor recorded at each point, a correspondence can be computed by examining the similarity between points from two input shapes. The similarities are often defined by a distance metric, forming an  $n_1 \times n_2$  distance matrix, where  $n_1$  and  $n_2$  are the point counts. The simplest way to arrive at a correspondence is the  $O(n_1 n_2)$  best matching, which greedily maps a point from one shape to a point on the other shape that has the most similar shape descriptor. Computing an optimal one-to-one correspondence which minimizes the total distance between matched shape descriptors is the bipartite matching problem and can be solved using the  $O(n^3)$  Hungarian algorithm [21], where  $n = \max(n_1, n_2)$ . These schemes can both be enhanced by a combination with iterative alignment, e.g., using thin-plate splines [1, 6].

Other optimization-based approaches also exist. Maciel and Costeira [17] formulate the matching problem via integer constrained minimization and solve a relaxed version of it using concave programming. Gold et al. [10] incorporate deformation parameters into their optimization criteria and find good suboptimal solutions via soft assign. Recently, Zheng and Doermann [35] improve upon initial correspondence computed from shape contexts by trying to maximize the preservation of binary neighborhood information. Most closely related to our work is the approach of Berg et al. [2], which also uses the QAP formulation but solves it using a two-step method. The last two algorithms [2, 35] both settle for gradient descent and converge to local minima.

The techniques mentioned so far are all applicable to point set correspondence. The more specialized contour matching problem can benefit greatly, quality-wise, under point ordering. Order-preserving contour matching typically involves expensive optimizations, but the reliance on a high-quality local shape descriptor is diminished. Liu et al. [16] resort to dynamic programming, which allows for feature skipping as well, and report excellent matching results. Scott and Nowak [24] also enforce point ordering and their dynamic programming algorithm is more efficient, but at the expense of a higher memory cost.

Instead of performing contour matching in the spatial domain, transform-based techniques have also gained some popularity, most notably spectral correspondence [13, 27] or modal matching [23]. In the resulting spectral domain, conventional methods can be applied to compute a corre-

spondence. The transforms used tend to reveal more global shape structures and can also obtain bending-invariant correspondences [4, 13]. Point correspondence without defining local shape descriptors is also possible. For example, Sederberg and Greenwood [26] present a physically based approach for correspondence of 2D polygons which tries to fit one polygon over another by optimizing some energy functional. Sebastian et al. [25] introduce the notion of an alignment curve and use a deformation-based symmetric edit distance to define contour similarity. Skeletal [32] and shock graphs [30] have also been proposed for 2D or 3D shape representations, where subsequent point correspondence can be computed via graph matching.

Finally, another class of techniques is based on the *minimum description length* (MDL) principle. Here the favoured point correspondence between a *group* of shapes is that which maximizes compactness (in an information theoretic sense) of the statistical shape model obtained using PCA on the corresponding points. MDL is not suitable for matching only a pair of contours (the focus of this paper) and does not explicitly include penalty when correspondence violates the proximity constraints.

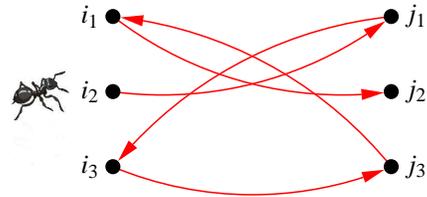
### 3. Review of ant colony optimization

Ant colony optimization is a metaheuristic used to find good solutions to NP-hard optimization problems, e.g., those related to routing, assignment (including QAP), and scheduling. A thorough introduction to ACO is given in the book of Dorigo [8], who pioneered this field.

#### 3.1 Overview of ACO framework

The inspiration for ACO comes from the ability of natural ant colonies to solve difficult problems, despite the simple behaviour of each individual member of the colony. This ability is reflected by the way ants search for food and communicate with each other. Initially, ants explore the surroundings of their nest in a random manner, a characteristic of their foraging behaviour. Whenever an ant finds a source of food, it returns to the nest leaving a trail of a chemical called *pheromone* on the ground. The purpose of this trail is to guide other ants to the food sources.

The ACO metaheuristic replicates some aspects of this behaviour. An artificial ant colony is seen as a system of distributed agents searching for the solution of an optimization problem. The problem is usually modelled with a graph and the solution search involves ants traversing this graph. Each traversal by an ant corresponds to a feasible solution, which is evaluated using an objective function of the problem at hand. The quality of the solution dictates pheromone deposition, whose accumulation on the traversed graph edges serves as the means for communication.



**Figure 2. Example of an ant traversal over a (3,3) bipartite graph modeling a correspondence problem. The path is  $(i_2 \rightarrow j_1 \rightarrow i_3 \rightarrow j_2 \rightarrow i_1 \rightarrow j_2)$  and the corresponding assignment  $\pi$  is:  $\pi(i_1) = j_2, \pi(i_2) = j_1, \pi(i_3) = j_3$ . The ant shown is not part of our graph model.**

In this work, we model the correspondence problem over a complete bipartite graph on the two point sets to be matched. As an ant traverses the graph, a path is formed. We define the obtained correspondence by those edges directed from one designated point set to the other and evaluate it using a criterion from the QAP formulation. Figure 2 shows a possible ant traversal and the correspondence formed. The details of our algorithm are given in Section 4.

#### 3.2 ACO metaheuristics

As an ant traverses the graph, its decision on where to go next are influenced by *heuristic information* and current pheromone deposition. For correspondence, the heuristic information will take into account both local shape descriptors and proximity (Section 4.2.4). After a number of ants have traversed the graph, known as one iteration of ACO, a certain amount of pheromone is also *evaporated* from all edges. Pheromone evaporation occurs in nature and in ACO, it can help the ants escape from local minima.

When examining the ACO algorithm, we see that, at first, ants will tend to freely explore the whole solution space, leading to many different solutions. However, over time, pheromones will accumulate only on edges that are part of those traversals favoured by the objective function; this causes the ants to gradually follow only a limited number of traversals. Moreover, heuristic information and pheromones have to be combined to guide an ant. The former is necessary to bias the ants to construct good traversals at the start, when the pheromones are set to random initial values. On the other hand, the pheromones are necessary for later iterations of the algorithm, when they reinforce the traversal of the graph edges that lead to good solutions.

### 4. ACO for shape correspondence

This section formulates the correspondence problem and describes how the ACO metaheuristic is applied.

#### 4.1. Problem formulation using QAP

Given two point sets  $I$  and  $J$ , the shape correspondence problem can be stated as finding a meaningful mapping from points of  $I$  to points of  $J$  which minimizes a given objective function. That is, we seek  $\pi^*$  such that

$$\pi^* = \operatorname{argmin}_{\pi} (\operatorname{OBJ}(\pi, I, J)),$$

where  $\operatorname{OBJ}$  is the objective or cost function which evaluates the matching  $\pi$  in relation to the shapes characterized by  $I$  and  $J$ , and  $\pi$  is a mapping such that  $\forall i \in I, \exists j \in J : \pi(i) = j$ . We also assume without loss of generality that  $|I| \leq |J|$ .

**The assignment problem (AP):** A common approach to shape correspondence is to extract a set of features for each point, referred to as the *shape descriptors*. Examples of shape descriptors include shape contexts [1], among many others [20]. Next, a distance measure between two shape descriptors has to be defined. The *assignment problem* or AP [24] seeks a correspondence which minimizes the sum of the distances between descriptors of points on one set and the descriptors of the corresponding points on the other set:

$$\operatorname{AP}(\pi, I, J) = \sum_{i \in I} \mathcal{D}_R(R_i, R_{\pi(i)}),$$

where  $R_i$  denotes the descriptor at point  $i$ , and  $\mathcal{D}_R$  is the shape descriptor distance measure. Here, the matching is constrained to be one-to-one (not necessarily onto). The optimal matching for this cost measure can be computed in cubic time with the Hungarian algorithm [21].

**Order preservation for contour matching:** Note that shape descriptors might not be the only factor one should take into account when evaluating a correspondence. One possible element pertaining to contour matching is order preservation [16, 24, 25], which follows from the observation that the vertices defining a contour are ordered. For example, the COPAP algorithm [24] solves the cyclic-order-preserving assignment problem via dynamic programming. It has been shown that order-preserving contour matching significantly improves the correspondence results [16, 24, 25]. However, it is unclear how it can be extended to other domains, e.g., for points residing on a 3D shape, for which a canonical point ordering is not available.

**Incorporating proximity information:** Another element for evaluating a correspondence between continuous shapes, e.g., contours or surfaces, is the preservation of quantitative neighborhood or *proximity* information. Namely, if point  $i$  from shape  $I$  and point  $j$  from shape  $J$  are matched, then a close-by neighbor  $i'$  of  $i$  on  $I$  should be matched with a point  $j'$  on  $J$  that is close to  $j$ . Berg et al. [2] refer to this as the minimization of distortions in a correspondence. This differs from the case of order preservation in that besides point ordering, we are also concerned with distances between point pairs on the same shape.

**QAP formulation:** When augmenting the shape descriptor  $R$  with proximity information, we obtain the general objective function of QAP,

$$\operatorname{QAP}(\pi, R, I, J) = (1 - \nu) \mathcal{S}(\pi, R, I, J) + \nu \mathcal{X}(\pi, I, J), \quad (1)$$

where the free parameter  $0 \leq \nu \leq 1$  weighs between the shape descriptor distance or similarity term  $\mathcal{S}(\cdot)$  and the proximity term  $\mathcal{X}(\cdot)$ . In our implementation, we define

$$\mathcal{S}(\pi, R, I, J) = \frac{1}{|I|} \cdot \sum_{i \in I} \left| 1 - e^{\frac{-\mathcal{D}_R(R_i, R_{\pi(i)})^2}{\sigma_R}} \right|,$$

where the use of the Gaussian is for normalization purposes, so that  $0 \leq \mathcal{S}(\pi, R, I, J) \leq 1$ . Note that other filter functions that are monotonically decreasing away from zero can also be used in place of the Gaussian. The proximity term

$$\mathcal{X}(\pi, I, J) = \frac{\sum_{i \neq i' \in I} \left[ e^{\frac{-\mathcal{D}_I(i, i')^2}{\sigma_I}} |\mathcal{D}_I(i, i') - \mathcal{D}_J(\pi(i), \pi(i'))| \right]}{|I| - 1/2}$$

also lies in  $[0, 1]$ , where  $\mathcal{D}_I$  and  $\mathcal{D}_J$  are distances, normalized to  $[0, 1]$ , between two points from  $I$  and  $J$ , respectively. For matching contours, these distances are geodesic distances normalized with respect to contour length. If the point sets are unorganized, we can resort to Euclidean distances normalized against the maximal pairwise distance in each set.

Moreover, proximity preservation is emphasized more in the local neighborhood of a given vertex and it diminishes for points that are far away from each other. Our way to accentuate proximity is to apply Gaussian weights to the proximities in the  $\mathcal{X}(\cdot)$  term. The Gaussian widths  $\sigma_R$  and  $\sigma_I$  applied to the descriptor distances and proximities are free parameters set by the user.

Finally, note that a great advantage of our QAP formulation is that it can be extended to any domain where distances between points can be computed; this includes the case of points residing on a 2-manifold, where  $\mathcal{D}_I$  and  $\mathcal{D}_J$  will be geodesic distances over the manifold. However, finding an optimal solution to QAP is NP-hard. Therefore, heuristic algorithms are necessary to obtain approximate solutions.

#### 4.2. Correspondence algorithm using ACO

It has been experimentally shown that the ACO meta-heuristic is one of the most successful approaches for solving structured real-life instances of the QAP [8]. Moreover, when incorporating proximity information, solving the correspondence problem can be viewed as solving a QAP, as we have shown in the last section. In this section, we describe a novel extension of the ACO framework, which has been used for solving assignment problems [18], to deal with the specific shape correspondence problem.

### 4.2.1 Graph model and correspondence extraction

We define the graph  $G = \{V, E\}$  that is to be traversed by the ants as a *complete, directed bipartite graph*. The set of vertices of this graph is composed of the two point sets to be matched, i.e.,  $V = I \cup J$ . The directed edges  $E$  fully connect the two point sets. The path that is determined by the traversal of an ant on  $G$  corresponds to a possible solution to the assignment (correspondence) problem. During such a traversal, two conceptually distinct tasks are performed. When an ant traverses a directed edge that connects a vertex in  $I$  to a vertex in  $J$ , an assignment from  $I$  to  $J$  is determined. On the other hand, when an ant traverses an edge from  $J$  to  $I$ , the order in which the vertices are assigned is determined.

An ant starts the graph traversal from a randomly selected vertex. It traverses edges until an assignment of each vertex in  $I$  is determined (recall that  $I \leq J$ ). The final assignment  $\pi$  is given by the edges from  $I$  to  $J$  chosen by the ant. Note that a correspondence obtained this way is not necessarily one-to-one, as opposed to the AP or COPAP matching formulations. In Section 4.3, we show how ACO is a flexible framework which allows other types of matching.

### 4.2.2 Path construction and evaluation

When traversing from a vertex  $i \in I$  to a vertex in  $J$ , the probability  $p_{ij}^k$  of an ant  $k$  choosing the edge that connects to vertex  $j \in J$  is given by

**Edge probability:** 
$$p_{ij}^k = \frac{\alpha \tau_{ij} + (1 - \alpha) \eta_{ij}}{\sum_{l \in N_i} [\alpha \tau_{il} + (1 - \alpha) \eta_{il}]} \quad (2)$$

where  $\tau_{ij}$  quantifies the pheromones accumulated on edge  $(i, j)$ ,  $\eta_{ij}$  indicates the desirability of traversing  $(i, j)$  based on heuristic information (defined in Section 4.2.4), and  $N_i = \{l \in V : (i, l) \in E\}$  is the immediate neighbourhood of vertex  $i$ . The parameter  $0 \leq \alpha \leq 1$  regulates the influence of pheromones over heuristic information. We can see that the choice of the traversed edge is stochastic, where the sum of probabilities  $\sum_{j \in N_i} p_{ij}^k = 1$ . Moreover, when traversing back from  $J$  to  $I$ , an edge pointing to any vertex in  $I$  that has not been visited yet can be chosen. Each edge from  $j$  to  $I$  has the same probability of being selected, and no pheromones or heuristic information are considered. This choice is certainly not unique and our framework is quite flexible in allowing for other variants, as we shall describe later in Section 4.3.

After  $m \geq 1$  ants have traversed the graph, completing an ACO iteration, the corresponding solutions are evaluated and the pheromones are accordingly updated. The cost of each solution, a correspondence  $\pi$  extracted from a path traversed by an ant, is given by Equation (1), where the distances between two shape descriptors and the geodesic distances between vertices on each contour are combined.

### 4.2.3 Pheromone updates

Pheromones are updated at the end of an ACO iteration. First, pheromones are evaporated at a constant *pheromone evaporation rate*  $\rho$ ,  $0 \leq \rho \leq 1$ ,

**Pheromone evaporation:** 
$$\tau_{ij} \leftarrow (1 - \rho) \tau_{ij}, \quad (3)$$

for all edges  $(i, j)$ . Next, new pheromone is deposited only on *edges that were traversed by the ants*,

**Pheromone deposition:** 
$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k, \quad (4)$$

where  $\Delta \tau_{ij}^k$  is the amount of pheromone that ant  $k$  has deposited on edge  $(i, j)$ . It is given as a constant  $\delta$  (a free parameter) divided by the correspondence cost defined above: the larger the cost, the less the amount of deposited pheromones. In addition, a minimum level of pheromones  $\tau_{min}$  is maintained on all edges to avoid completely eliminating certain traversals during the ants' exploration.

### 4.2.4 Heuristic information for ant traversal

What remains to be determined now is how to compute the heuristic information  $\eta_{ij}$  in (2). Naturally, our proposed heuristic is a combination of descriptor distance and the geodesic distance between vertices on the same shape, in a manner similar to the QAP cost in (1). Specifically,

$$\eta_{ij} = \left( e^{\frac{-\mathcal{D}_R(R_i, R_j)^2}{\sigma_R}} \right) \times \left( 1 - e^{\frac{-\mathcal{D}_I(i, i')^2}{\sigma_I} \left| \mathcal{D}_I(i, i') - \mathcal{D}_J(j, \pi(i')) \right|} \right) \times \left( 1 - e^{\frac{-\mathcal{D}_I(i, i'')^2}{\sigma_I} \left| \mathcal{D}_I(i, i'') - \mathcal{D}_J(j, \pi(i'')) \right|} \right)$$

where  $i'$  and  $i''$  are respectively the last and second last vertices visited in  $I$ , and  $\pi$  is already defined for all the previously visited vertices. The possible distance values  $\mathcal{D}_R$  (for shape descriptor  $R$ ),  $\mathcal{D}_I$ , and  $\mathcal{D}_J$  are all normalized to  $[0, 1]$ , as before. Therefore, the heuristic information has the maximum value of 1 when all the distances are very small (zero), and lower values when all the distances are large. In this way, assignments which relate points with similar descriptors and encourage preservation of proximity information are favored by the heuristic.

Note that we can consider more than two last visited vertices in the heuristic. However, since the previous assignments also take into consideration of proximity, information from the last two vertices appears to be sufficient to maintain the proximity between the assigned vertices.

ACO Parameter	Symbol	Value
Number of ants	$m$	1
Number of iterations	$T$	1000
Influence of pheromones	$\alpha$	0.3
Pheromone evaporation rate	$\rho$	0.1
Pheromone deposition constant	$\delta$	0.01
Initial pheromone levels	$\tau_0$	1
Minimum pheromone levels	$\tau_{\min}$	$0.1 \cdot \frac{1}{ I }$
Influence of proximity	$\nu$	0.7
Gaussian width in $\mathcal{X}$	$\sigma_I$	$0.1 \cdot I_{\max}$
Gaussian width in $\mathcal{S}$	$\sigma_R$	$0.1 \cdot R_{\max}$

**Table 1. Parameters used in our ACO algorithm and their chosen values.**

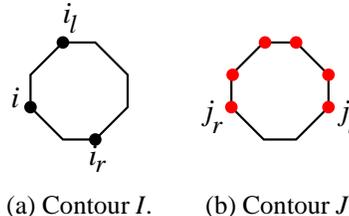
#### 4.2.5 List of ACO parameters and values

In Table 1, we list all the parameters used in our ACO shape matching algorithm and the respective values that are chosen in our experiments. Note that the auxiliary parameters  $I_{\max}$  and  $R_{\max}$  are data dependent and they are respectively set to be the maximum proximity (in  $I$ ) and the maximum descriptor distance. Although appropriate tuning of the parameters is necessary for a good performance of the algorithm, it will be shown that the ACO approach is sufficiently robust to work on a wide range of contours from different classes using the same set of parameter values.

#### 4.3. Extensions to basic ACO framework

**Order preservation:** It is quite straightforward to modify our basic ACO algorithm described so far so that only correspondences that preserve the ordering of the contour vertices are constructed. Basically, when the algorithm is computing the probability of matching a vertex  $i \in I$  to a certain vertex  $j \in J$ ,  $j$  is only assigned a non-zero probability if it lies in a *valid interval* of vertices that preserve the ordering of the matching. This interval is computed as follows. Firstly, vertices  $i_l$  and  $i_r$  are found in the first contour, which are respectively the left and right neighbors of vertex  $i$  that already have defined matchings  $\pi(i_l) = j_l$  and  $\pi(i_r) = j_r$ . Next,  $j$  lies in the valid interval if it is contained between  $j_l$  and  $j_r$ , taking into consideration the cycle of the contour; see Figure 3. It is assumed that both contours are consistently ordered along the same direction; if this were not the case, we could always reverse the vertex ordering of one contour and perform a second matching.

**Open curves:** These are trivial to handle by ACO: the only distinction from contour matching is related to the determination of the valid vertex range for order preservation. With open curves, their end points serve as hard boundaries.



**Figure 3. Valid range of vertices for order-preserving contour matching. If the matching  $\pi$  has been partially constructed with  $\pi(i_l) = j_l$  and  $\pi(i_r) = j_r$ . Then the set of vertices on contour  $J$  (b) that are in the valid range for matching  $i$  on contour  $I$  (a) are marked in red.**

**Flexible matching paradigm:** While AP and COPAP enforce one-to-one correspondence, our ACO framework can be easily tuned to allow for one-to-one, one-to-many, many-to-one, or many-to-many matchings. All that is needed is to place hard constraints or assign appropriate probabilities to allow/disallow or encourage/discourage certain edges to be traversed by an ant. For example, we allow many-to-one matching by assigning non-zero probabilities to edges pointing from  $J$  towards already assigned vertices in  $I$ .

#### 4.4. Pseudocode for ACO correspondence

Figures 4, 5, 6, and 7 present the pseudocode for our ACO-based shape matching algorithm described in the previous sections, where order preservation is enforced to emphasize our focus on contour correspondence. The main function is **ACO-ShapeMatching**, given in Figure 4, and it calls several helper functions given in subsequent figures.

### 5. Experimental results

To demonstrate the capabilities of our method, a set of experiments were performed on the Brown dataset [28] and the results are presented here. The Brown dataset is frequently used in the computer vision community as test data for shape analysis algorithms. It consists of the silhouette images of objects such as hands, humans, fish, airplanes, etc., some of which were captured under occlusion. We are using the collection from this dataset that is composed of 9 object classes with 11 images per class. In our experiments, a contour of 70 vertices is extracted for each image.

The original (rotation-variant) shape context [1] is used as the descriptor in most of the experiments. The similarity between two shape context histograms is given by the  $\chi^2$ -distance. We primarily compare results we obtain with those of COPAP [24], which is a recent and one of the most successful contour matching algorithms.

```

1: ACO-ShapeMatching( $I, J$ ):
2:    $G \leftarrow \text{initGraph}(I, J)$ 
3:    $\text{BestMatching} \leftarrow \emptyset$ 
4:    $\text{BestCost} \leftarrow \infty$ 
5:   for  $i \leftarrow 1$  to  $T$  do
6:      $\text{Matchings} \leftarrow \emptyset$ 
7:     for  $j \leftarrow 1$  to  $m$  do
8:        $M \leftarrow \text{constructMatching}(G)$ 
9:        $C \leftarrow$  Cost of matching  $M$  according to Equation (1)
10:       $\text{Matchings} \leftarrow \text{Matchings} \cup \{M, C\}$ 
11:      if  $C < \text{BestCost}$  then
12:         $\text{BestMatching} \leftarrow M$ 
13:         $\text{BestCost} \leftarrow C$ 
14:      end if
15:    end for
16:    updatePheromones( $G, \text{Matchings}$ )
17:  end for
18:  return  $\{\text{BestMatching}, \text{BestCost}\}$ 

```

**Figure 4.** The main function for our ACO-based shape correspondence algorithm.

```

1: initGraph( $I, J$ )
2:    $G \leftarrow$  A complete, directed, bipartite graph between  $I$  and  $J$ 
3:   for each edge  $e$  in  $\{I \rightarrow J\} \subset E(G)$  do
4:      $e_{pher} \leftarrow \tau_0$ 
5:   end for
6:   for each edge  $e$  in  $\{J \rightarrow I\} \subset E(G)$  do
7:      $e_{pher} \leftarrow 0$ 
8:   end for
9:   return  $G$ 

```

**Figure 5.** Graph model initialization.

```

1: constructMatching( $G$ )
2:    $M \leftarrow \emptyset$ 
3:    $i \leftarrow$  Randomly chosen from the vertices in  $I$ 
4:   while there are unmatched vertices in  $I$  do
5:     for each vertex  $j$  in  $J$  do
6:       /* for order preservation */
7:       if  $j$  is in the set of valid  $J$  vertices then
8:          $P[j] \leftarrow \frac{\alpha \tau_{ij} + (1-\alpha) \eta_{ij}}{\sum_{l \in N} \alpha \tau_{il} + (1-\alpha) \eta_{il}}$ 
9:       else
10:         $P[j] \leftarrow 0$ 
11:      end if
12:    end for
13:     $j_{\text{choice}} \leftarrow$  Probabilistically chosen vertex in  $J$ 
      according to probabilities  $P$ 
14:     $M \leftarrow M \cup \{i, j_{\text{choice}}\}$ 
15:     $i \leftarrow$  Randomly chosen from the set of available
      vertices in  $I$ 
16:  end while
17:  return  $M$ 

```

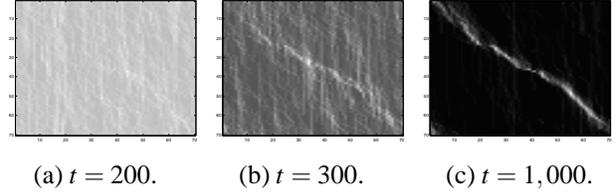
**Figure 6.** Matching construction function.

```

1: updatePheromones( $G, \text{Matchings}$ )
2:   /* Evaporate pheromone */
3:   for each edge  $e$  in  $\{I \rightarrow J\} \subset E(G)$  do
4:      $e_{pher} \leftarrow (1-\rho)e_{pher}$ 
5:   end for
6:   /* Add new pheromone */
7:   for each  $\{M, C\}$  in  $\text{Matchings}$  do
8:      $\Delta_{pher} \leftarrow \delta/C$ 
9:     for each edge  $e$  in  $\{I \rightarrow J\}$  in matching  $M$  do
10:       $e_{pher} \leftarrow e_{pher} + \Delta_{pher}$ 
11:    end for
12:   end for
13:   /* To ensure minimum pheromone level */
14:   for each edge  $e$  in  $\{I \rightarrow J\} \subset E(G)$  do
15:     if  $e_{pher} < \tau_{\min}$  then
16:        $e_{pher} \leftarrow \tau_{\min}$ 
17:     end if
18:   end for

```

**Figure 7.** Pheromone update function.

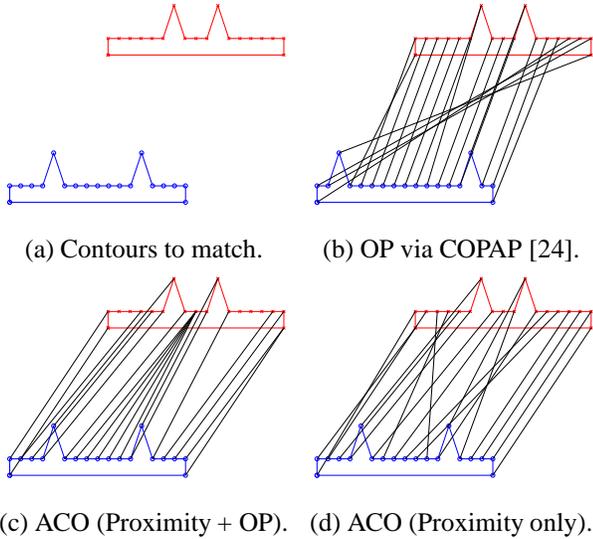


(a)  $t = 200$ . (b)  $t = 300$ . (c)  $t = 1,000$ .

**Figure 8.** Pheromone deposit during an execution of our ACO-based correspondence algorithm after  $t = 200, 300$ , and  $1,000$  iterations.

**ACO pheromone deposition:** Firstly, the proposed method is examined from the view of the ACO metaheuristic. Figure 8 presents several  $70 \times 70$  matrices representing pheromone deposition during one execution of the algorithm, when the correspondence for two similar contours is computed. Each pixel at position  $(i, j)$  represents the pheromone deposited at edge  $(i, j)$ , where lighter colors indicate larger numbers. It can be seen that all edges possess similar levels of pheromone at the beginning of the computation. However, as the iterations advance, pheromone concentrates on certain edges, leading to a configuration where only a few traversals are favored during the computation. These edges represent the assignments that heuristically provide the best correspondences.

**Order preservation (OP) vs. proximity consideration:** Next, the quality of the computed correspondences is addressed. In order to verify that the incorporation of proximity information has a significant contribution in the quality of the matchings, Figures 1 and 9 present two cases where the matchings are computed with and without the addition



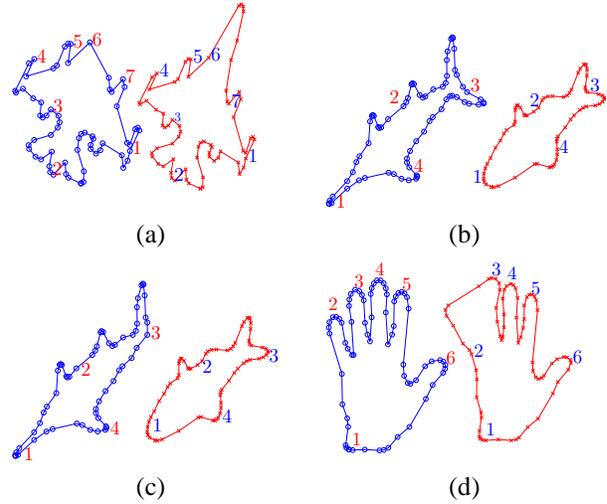
**Figure 9. Effect of incorporating proximity information when matching two contours (a) with the same features but under stretching. (b) Results from order preservation (OP) only. (c) ACO with proximity and OP. (d) ACO with proximity information and without OP.**

of proximity consideration. Additionally, Figure 9 presents a case where proximity information is used, but no order preservation is imposed. The descriptor used in these two examples is the angle at each vertex. The results indicate that the incorporation of proximity information provides matchings that tend to be more intuitive than those computed when only order preservation is imposed, especially for cases where there is a lack of salient features (Figure 1) or when nonuniform stretching occurs (Figure 9).

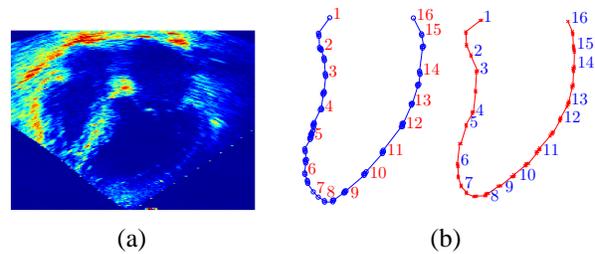
**Handling of occlusion or missing parts:** Figure 10 shows the correspondences computed for models that have missing or occluded parts. The first three cases were compared before by Sebastian et al. [25], which failed in case (c). To avoid cluttering in the figures, we only present the matching of relevant feature points as marked. It can be seen that our algorithm computes the correct matchings for all the model pairs, not being significantly affected by changes in the structure of the shapes, such as the missing part happening from (b) to (c) or the occlusion in case (d).

**Handling of open contours:** The correct correspondence for two open contours is computed by our ACO algorithm and is presented in Figure 11(b). The two left-ventricular contours were extracted from an echocardiography image sequence, with one frame shown in Figure 11(a).

**Evaluation against ground-truth correspondence:** In Table 2, we present differences between the correspondences



**Figure 10. Matchings computed by ACO for contours with occlusion or structure change.**



**Figure 11. Matching computed by ACO for an open contour of a left ventricle.**

obtained by our algorithm and the ground truth, collected over the whole Brown dataset. The ground truth is provided by a human user for a selected set of feature points along the contours. Note that this is done for all the shape classes in the database except for one of the “fgen” shapes for which the ground truth has been hard to establish. Next, the geodesic distances along the contour between the ground-truth matching points and the matchings returned by the algorithm are summed up; such a scoring scheme has been proposed by Karlsson and Ericsson [14]. Obviously, the smaller the distance measure obtained, the closer the computed matching is to the ground truth.

Each number in Table 2 is a sum of the distances collected for all contour pairs from a particular class. Our algorithm is compared to the COPAP algorithm [24], and to AP, which is computed by the Hungarian algorithm. It can be inferred from the results that our algorithm generally provides matchings that are closer to the ground truth

Shape class	Hungarian	COPAP	ACO
Airplanes	223.16	32.55	13.02
Fish	56.85	21.67	22.80
Four-legged	235.57	32.58	25.48
Hands	375.94	94.86	121.95
Humans	482.27	53.75	20.95
Rabbits	190.01	80.01	53.44
Stingrays	30.55	5.88	5.16
Tools	204.36	35.29	22.48

Table 2. Deviation from ground truth.

when compared to both algorithms; this is more so for the airplane, human, and tool classes. It is worth noting that since the local shape descriptor we use is not rotationally invariant, the correspondence tests are conducted after the shapes have been pre-aligned.

**Shape retrieval:** We conduct a retrieval experiment on a collection of 25 images chosen from the Brown dataset [24]; this has been a fairly standard test performed by several others. Each shape is matched to all the others in the collection, and the matching cost is used as a measure of similarity between the contours for retrieval. Next, the three most similar contours are obtained. The number of first, second, and third matches that belong to the same class are counted. The result obtained for COPAP is, respectively, 21, 19, and 17. For our algorithm, we obtained 21, 20, and 17. It should be noted that these two tests were conducted using the original shape context as the descriptor, while the better scores reported in [24] used a rotationally invariant version of it.

**Test on unorganized point sets:** The same shape retrieval experiment described above has also been performed on the 25-image Brown dataset where the models are now regarded as general (unorganized) point sets and not considering the ordering of the contours. The results obtained were 21, 19, and 18, reinforcing that just the incorporation of proximity information can provide high quality results.

**Parameters:** It is important to be aware that all the results presented here for our algorithm were obtained with the same set of parameters (shown in Table 1), which confirms the robustness of our ACO approach.

**Timing:** Figure 12 plots the execution time required by our algorithm (1,000 ACO iterations), in comparison to COPAP and Hungarian, when contour sizes increase. Although our algorithm is comparable to the other two for small contours, it clearly scales better as vertex counts grow. The memory requirements for COPAP also increase significantly with the contour size, while those of ACO increase linearly.

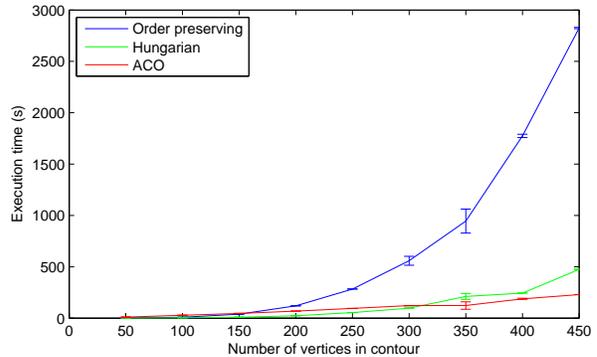


Figure 12. Execution time comparison between Hungarian, COPAP, and ACO.

## 6. Conclusions and Future Work

We formulate shape correspondence as a Quadratic Assignment Problem (QAP), incorporating proximity information into the point matching objective function. We also propose the first Ant Colony Optimization (ACO) algorithm directly aimed at solving the point and contour correspondence problems, which are difficult problems when the QAP formulation is adopted. The advantage of incorporating proximity and the effectiveness of the proposed method have been verified with a set of experiments. Qualitative and quantitative results show that the correspondences obtained by our ACO algorithm are at least comparable to those obtained by the best alternative methods, e.g., [24, 25]. In several cases, we have demonstrated clear advantages offered by our approach. In addition, our algorithm also has the advantage that its resource requirements scale moderately for contours of increasing size.

Although the descriptor used in our experiments did not allow to compute matchings for planar shapes with articulated deformations, the original shape context can be extended using geodesic neighborhood information [12]; this can be incorporated into our ACO framework easily, as well as any shape descriptor that allows to compute the similarity between vertices on two shapes.

Moreover, the proposed QAP-based ACO framework is quite flexible in several aspects: (1) Hard constraints can be incorporated by simply restricting the ant traversals to a reduced set of edges, as is done for enforcing order preservation; (2) Soft constraints can also be added by assigning different probabilities to different edges. For example, order preservation can be *favoured* rather than *enforced* through assigning non-zero and non-unit probabilities to different edges. The classical order preservation constraint can then be thought of as a special case of this new formulation; (3) It

can be extended to surface matching by replacing proximities measured along a contour with proximities measured over a surface (e.g., using geodesic distance), and by extracting surface shape descriptors for the vertices (e.g. 3D shape context [15] or local surface curvature [9]); (4) It can be applied to point clouds rather than ordered vertices on contours or meshes by also modifying the proximity computation (e.g., using Euclidean distance).

Our future work includes the aforementioned extensions to the matching of 3D meshes and point clouds. Moreover, experiments with other shape descriptors and comparisons with alternative techniques are also planned in order to further confirm the effectiveness of our ACO algorithm.

## References

- [1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape context. *PAMI*, pages 509–522, 2001.
- [2] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR)*, pages 26–33, 2005.
- [3] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *PAMI*, 14(2):239–256, 1992.
- [4] A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel. Matching two-dimensional articulated shapes using generalized multidimensional scaling. In *Proc. Conf. on Articulated Motion and Deformable Objects*, 2006.
- [5] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. In *Computer Graphics Forum*, pages 223–232, 2003.
- [6] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89:114–141, 2003.
- [7] T. F. Cootes, D. Cooper, C. J. Taylor, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
- [8] M. Dorigo and T. Stutzle. *Ant Colony Optimization*. The MIT Press, July 2004.
- [9] T. Gatzke, C. Grimm, M. Garland, and S. Zelinka. Curvature maps for local shape comparison. In *Proc. Shape Modeling International*, 2005.
- [10] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjølness. New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern Recognition*, 31(8):1019–1031, 1998.
- [11] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover’s distance. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR)*, pages 220–227, 2004.
- [12] V. Jain and H. Zhang. Robust 2d shape correspondence using geodesic shape context. In *Proc. of Pacific Graphics*, pages 121–124, 2005.
- [13] V. Jain, H. Zhang, and O. van Kaick. Non-rigid spectral correspondence of triangle meshes. *Int. J. on Shape Modeling*, (to appear) 2007.
- [14] J. Karlsson and A. Ericsson. A ground truth correspondence measure for benchmarking. In *Proc. Int. Conf. on Pat. Rec. (ICPR)*, volume 3, pages 568–573, 2006.
- [15] M. Kortgen, G.-J. Park, M. Novotni, and R. Klein. 3D shape matching with 3D shape contexts. In *Central European Seminar on Computer Graphics*, 2003.
- [16] L. Liu, G. Wang, B. Zhang, B. Guo, and H.-Y. Shum. Perceptually based approach for planar shape morphing. In *Proc. of Pacific Graphics*, pages 111–120, 2004.
- [17] J. Maciel and J. P. Costeira. A global solution to sparse correspondence problems. *PAMI*, 25(2):187–199, 2003.
- [18] V. Maniezzo and A. Colomi. The Ant System applied to the quadratic assignment problem. *IEEE Trans. on Knowledge and Data Engineering*, 11(5):769–778, 1999.
- [19] D. Meyers, S. Skinner, and K. Sloan. Surfaces from contours. *ACM Trans. on Graphics*, 11(3):228–258, 1992.
- [20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.
- [21] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1982.
- [22] P. Pardalos, F. Rendl, and H. Wolkowicz. The quadratic assignment problem: a survey and recent developments. In P. Pardalos and H. Wolkowicz, editors, *Quadratic assignment and related problems*, pages 1–42. 1994.
- [23] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *PAMI*, 17(6):545–561, 1995.
- [24] C. Scott and R. D. Nowak. Robust contour matching via the order-preserving assignment problem. *IEEE Trans. on Image Processing*, 15(7):1831–1838, 2006.
- [25] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *PAMI*, 25(1):116–124, 2003.
- [26] T. W. Sederberg and E. Greenwood. A physically based approach to 2-D shape blending. In *Proc. SIGGRAPH*, pages 25–34, 1992.
- [27] L. S. Shapiro and J. M. Brady. Feature based correspondence: An eigenvector approach. *Image and Vision Computing*, 10(5):283–288, 1992.
- [28] D. Sharvit, J. Chan, H. Tek, and B. Kimia. Symmetry-based indexing of image databases. *J. Vis. Commun. Image Represent.*, 9(4):366–380, 1998.
- [29] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Proc. IEEE Conf. on Shape Modeling and Applications*, pages 167–178, 2004.
- [30] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *Int. J. on Comp. Vis. (IJCV)*, 35(1):13–32, 1999.
- [31] T. Stützle and M. Dorigo. ACO algorithms for the quadratic assignment problem. *New ideas in optimization*, pages 33–50, 1999.
- [32] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proc. IEEE Conf. on Shape Modeling and Applications*, pages 130–142, 2003.
- [33] T. Tangelder and R. Veltkamp. A survey of content based 3D shape retrieval methods. In *Proc. IEEE Conf. on Shape Modeling and Applications*, pages 145–156, 2004.
- [34] R. C. Veltkamp and M. Hagedoorn. State of the art in shape matching. pages 87–119, 2001.
- [35] Y. Zheng and D. Doermann. Robust point matching for non-rigid shapes by preserving local neighborhood structures. *PAMI*, 28(4):643–649, 2006.