Contour Correspondence via Ant Colony Optimization

Oliver van Kaick — GrUVi Lab
Ghassan Hamarneh — MIAL
Hao Zhang — GrUVi Lab
Paul Wighton — MCL

School of Computing Science
Simon Fraser University

SFU

gruvi graphics + usability + visualization

MIAL
Medical Image Analysis Lab

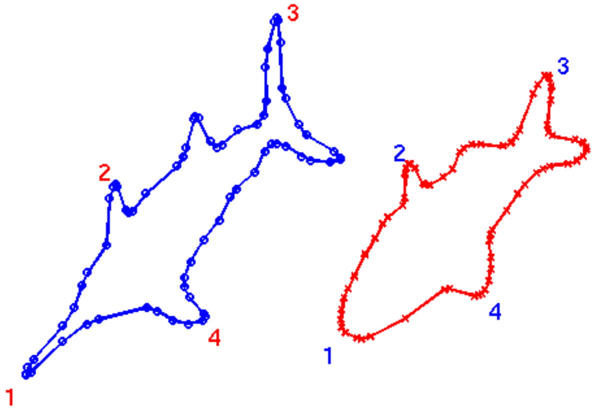
Outline

Outline

Introduction

- Shape correspondence
  - Finding a meaningful matching between two shapes

- Shape correspondence
  - Finding a meaningful matching between two shapes
- Focus: 2D contours



Introduction

- Applications in
  - Computer graphics (shape analysis, morphing, and animation)
  - Computer vision (object tracking, recognition, and retrieval)
  - Medical computing (statistical shape modeling and analysis of anatomical structures)

  - 3D shape matching and retrieval (Chen et al., 2003)

Solving contour correspondence

Common approach

1. Select feature points
2. Compute shape descriptors
3. Extract a matching

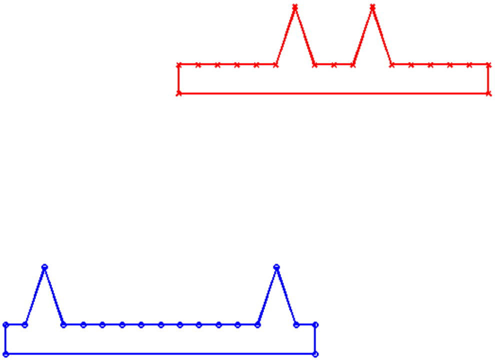
Solving contour correspondence

Common approach

1. Select feature points
2. Compute shape descriptors
3. Extract a matching
    - Greedy best matching
    - Bipartite matching
    - Iterative closest point (ICP) scheme
    - Dynamic programming under point ordering

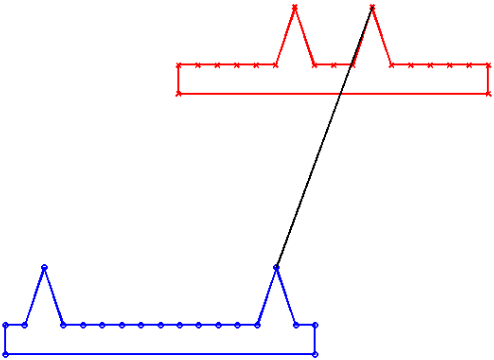
Incorporating proximity information

- Most algorithms do not consider proximity information
  Proximity: if two points are close on one shape, their corresponding points in the second shape should also be close



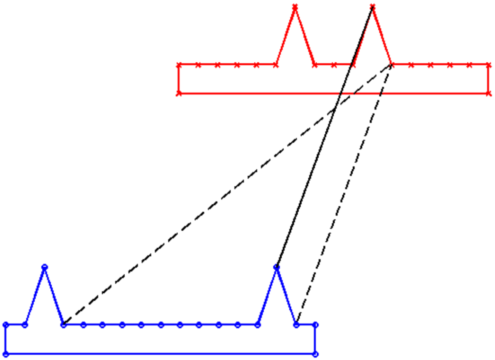
Incorporating proximity information

- Most algorithms do not consider proximity information
  Proximity: if two points are close on one shape, their corresponding points in the second shape should also be close



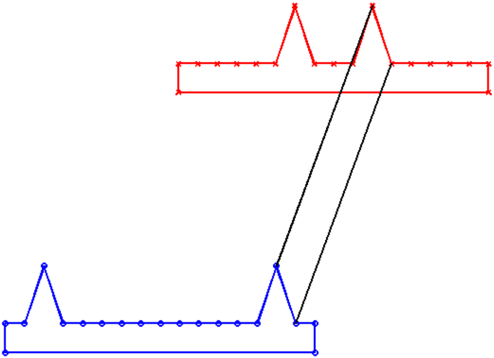
Incorporating proximity information

- Most algorithms do not consider proximity information
  Proximity: if two points are close on one shape, their corresponding points in the second shape should also be close

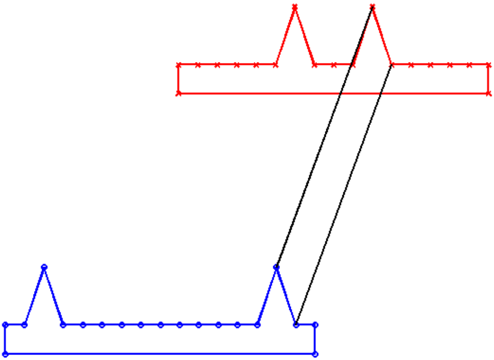


Incorporating proximity information

- Most algorithms do not consider proximity information
  Proximity: if two points are close on one shape, their corresponding points in the second shape should also be close



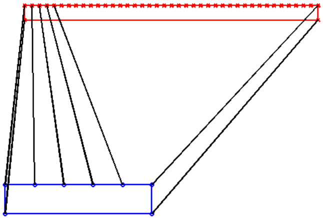
- Most algorithms do not consider proximity information
  Proximity: if two points are close on one shape, their corresponding points in the second shape should also be close

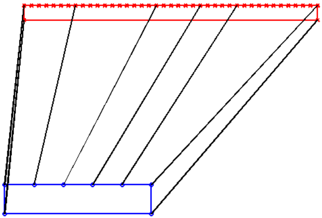


- Better handling of missing parts or a lack of salient features

- Take advantage of the vertex ordering (contours)

Enforcing **order preservation** $\neq$ Enforcing **proximity**





QAP

- When **proximity** is incorporated, we can formulate point correspondence via the **Quadratic Assignment Problem (QAP)**
- **QAP** is one of the **most difficult** optimization problems
- **Ant Colony Optimization (ACO)** has had great success in solving this problem

Contributions

- We formulate the general point correspondence problem in terms of **QAP** incorporating **proximity information**
- We propose the **first ACO algorithm** to compute the matching
- Applicable to **contours** and **unorganized 2D point sets**
- We extend the framework to enforce **order preservation** (contours)

Outline

Related work on correspondence

- Matching shape descriptors
  - 2D shape matching
    - Bipartite matching solved using the Hungarian algorithm
    - Integer constrained minimization (Maciel and Costeira, 2003)
    - Soft assign algorithm (Gold et al., 1998)
    - Preservation of binary neighborhood information (Zheng and Doermann, 2006)
    - QAP formulation (Berg et al., 2005)
  - Contour correspondence
    - Order preservation and dynamic programming (Liu et al., 2004, Scott and Nowak, 2006)

Related work on correspondence

- Matching shape descriptors
  - 2D shape matching
    - Bipartite matching solved using the Hungarian algorithm
    - Integer constrained minimization (Maciel and Costeira, 2003)
    - Soft assign algorithm (Gold et al., 1998)
    - Preservation of binary neighborhood information (Zheng and Doermann, 2006)
    - QAP formulation (Berg et al., 2005)
  - Contour correspondence
    - Order preservation and dynamic programming (Liu et al., 2004, Scott and Nowak, 2006)

Related work on correspondence

- Matching shape descriptors
  - 2D shape matching
    - Bipartite matching solved using the Hungarian algorithm
    - Integer constrained minimization (Maciel and Costeira, 2003)
    - Soft assign algorithm (Gold et al., 1998)
    - Preservation of binary neighborhood information (Zheng and Doermann, 2006)
    - QAP formulation (Berg et al., 2005)
  - Contour correspondence
    - Order preservation and dynamic programming (Liu et al., 2004, Scott and Nowak, 2006)

Related work on correspondence

- Without using local descriptors:
  - Physically-based approach (Sederberg and Greenwood, 1992)
  - Pattern matching in the Gaussian map (Tal and Elber, 1999)
  - Deformation-based edit distance (Sebastian et al., 2003)
  - Skeletal and shock graphs (Sundar et al., 2003, Siddiqi et al., 1999)
- Transform-based techniques (Shapiro and Brady, 1992, Sclaroff and Pentland, 1995, Bronstein et al., 2006, Jain et al., 2007)
- Group correspondence
  - Minimum Description Length (MDL) principle (Davies et al., 2002)

Outline

ACO facts

- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
    - Individual ants have a simple behavior
    - The ant colony can solve difficult problems

ACO facts

- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)

ACO facts

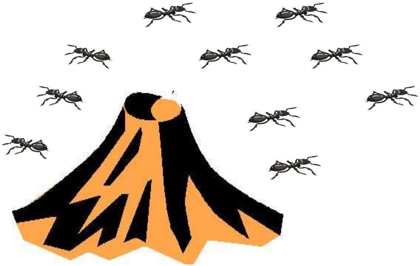
- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)

ACO facts

- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)

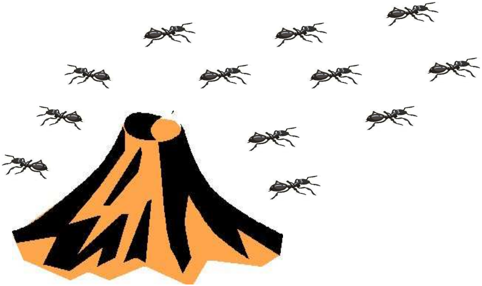
ACO facts

- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)



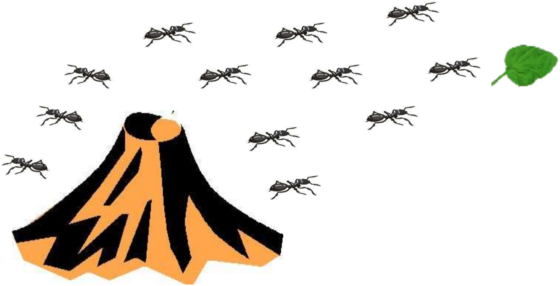
ACO facts

- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)

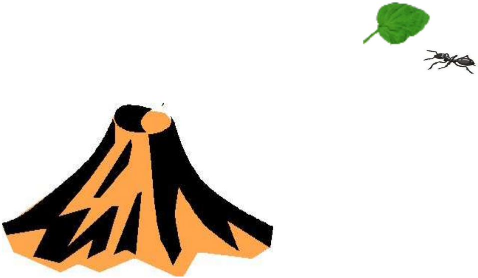


ACO facts

- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)



- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)

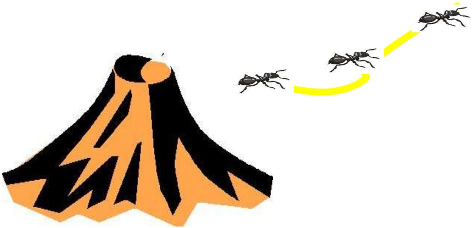


ACO facts

- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)

ACO facts

- Finds good solutions to NP-hard problems (Dorigo et al., 1996)
- Routing, assignment, and scheduling
- Inspiration from nature
  - Individual ants have a simple behavior
  - The ant colony can solve difficult problems
- Ant characteristics
  - Foraging behavior (search)
  - *Pheromone trail* (communication)



- Problem modeled with a graph
- The solution search involves ants traversing this graph

Review of the ACO framework

- Problem modeled with a graph
- The solution search involves ants traversing this graph

- ACO metaheuristic:

  For each iteration:

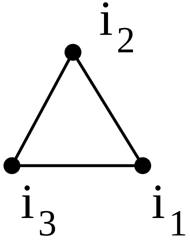
  1. Traverse the graph (construct solution)
     - Heuristic information and pheromones
  2. Evaluate solution
     - Objective function
  3. Deposit pheromones on the edges of the graph
     - Quality of the solution

Advantages of ACO

- Probabilistic approach
- Heuristic information
- Escape from bad local minima
- Parallelizable

Outline

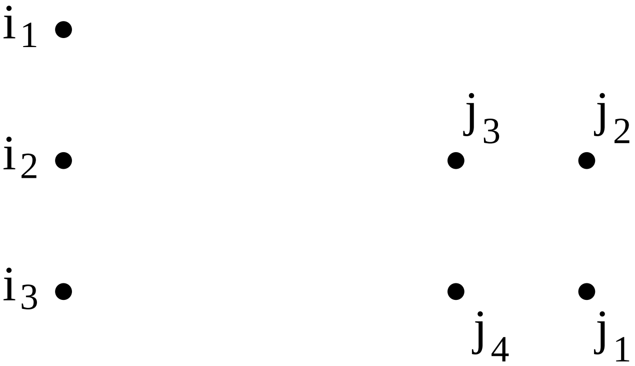
ACO for shape correspondence



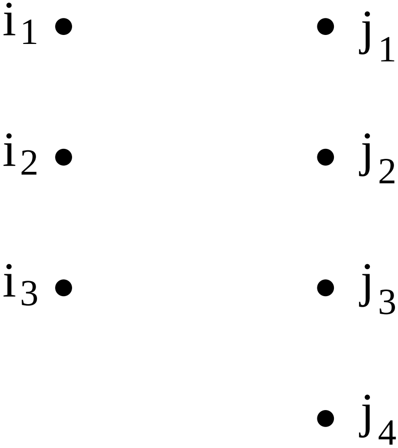
Shapes to be matched

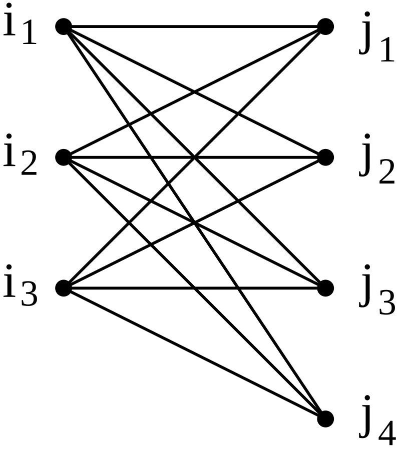

i_2
i_3
i_1
j_3
j_2
j_4
j_1

ACO for shape correspondence

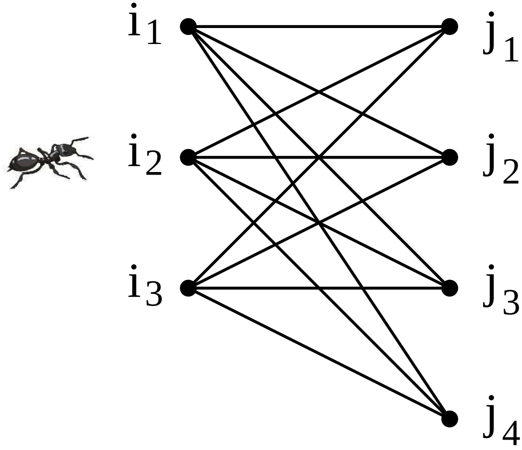


ACO for shape correspondence



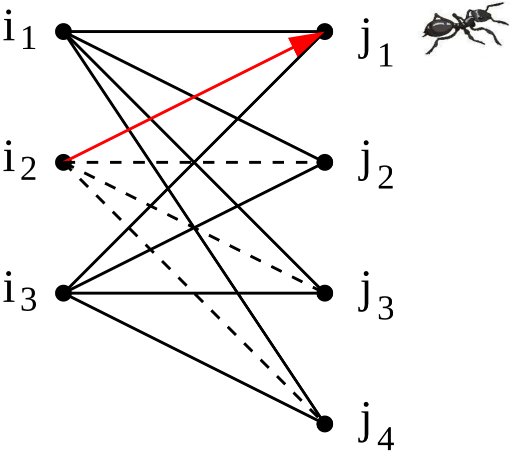


ACO graph



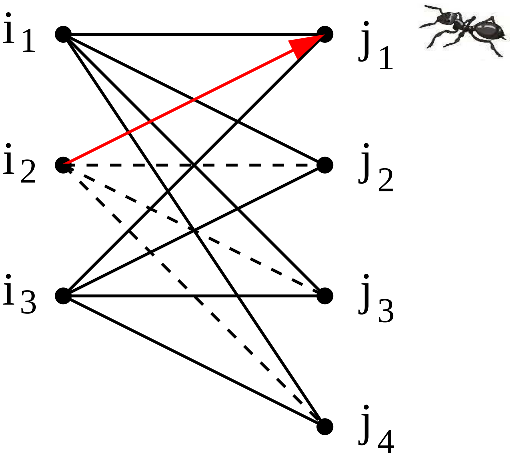
Iteration start

ACO for shape correspondence

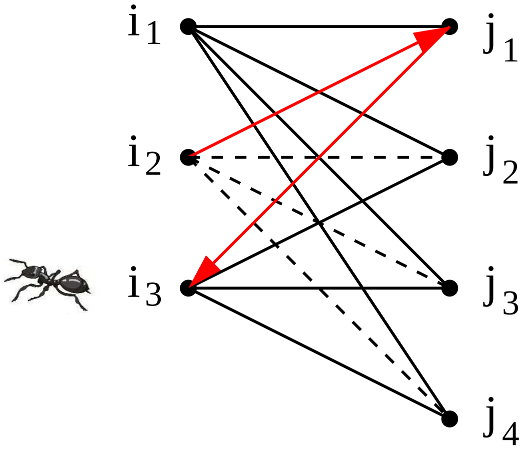


$$\pi(i_2) = j_1$$

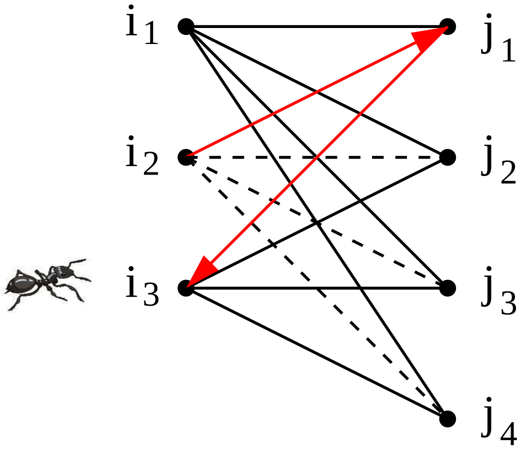
ACO for shape correspondence



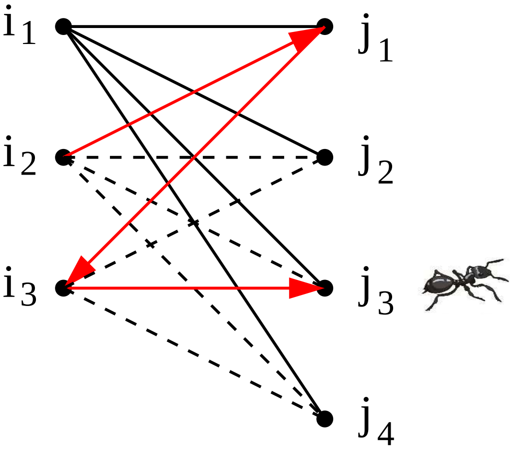
ACO for shape correspondence



ACO for shape correspondence

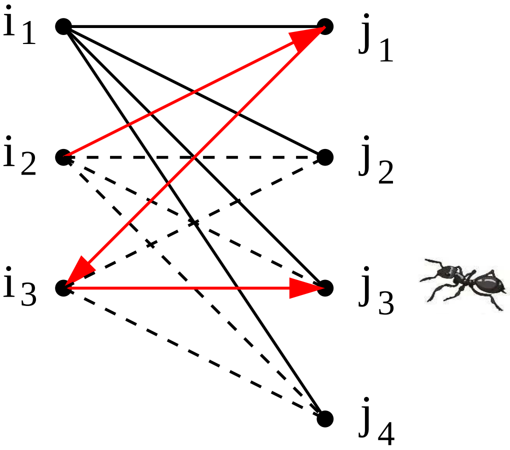


ACO for shape correspondence

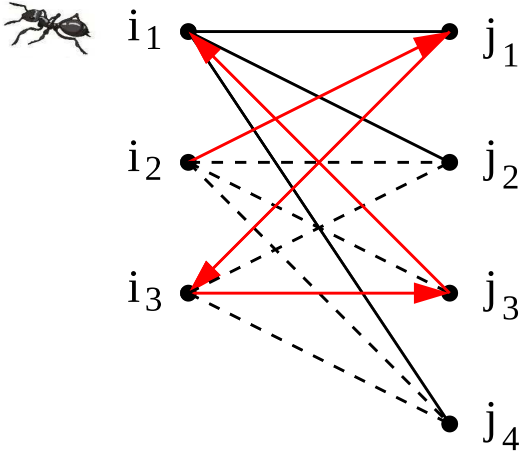


$$\pi(i_3) = j_3$$

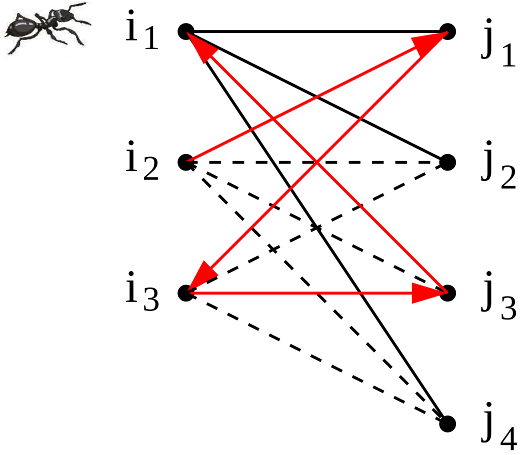
ACO for shape correspondence



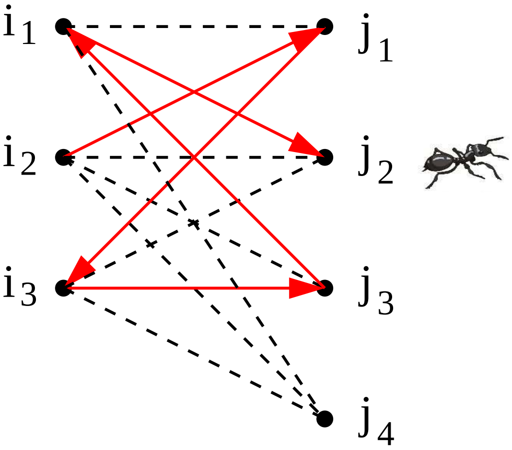
ACO for shape correspondence



ACO for shape correspondence

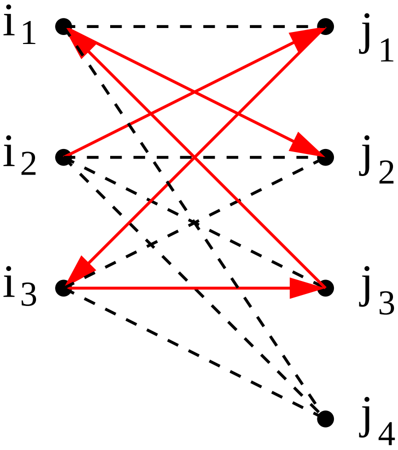


ACO for shape correspondence



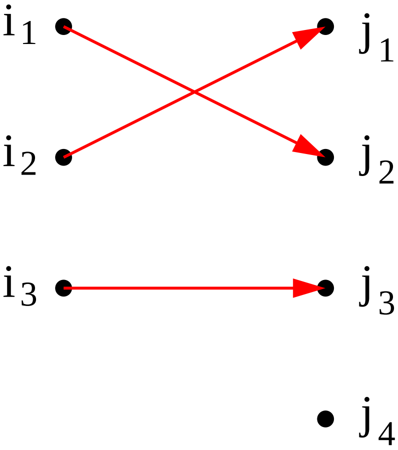
$$\pi(i_1) = j_2$$

ACO for shape correspondence



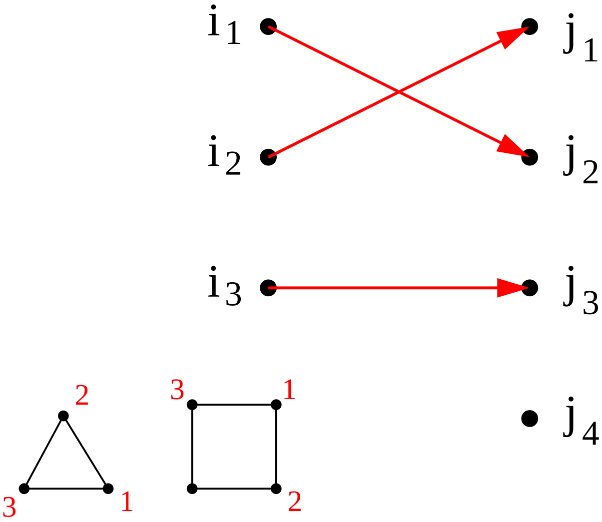
Iteration end

ACO for shape correspondence

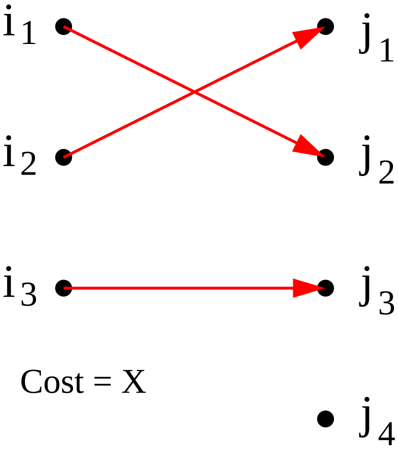


Correspondence obtained

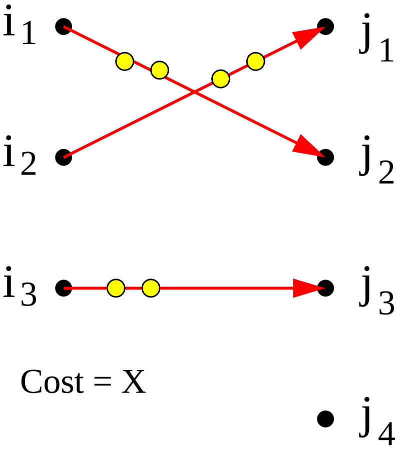
ACO for shape correspondence



ACO for shape correspondence



Cost is computed



Pheromone is deposited

ACO for shape correspondence

- For a fixed number of iterations:
  1. Traverse the graph
     - Heuristic information and pheromones
  2. Evaluate solution
     - Objective function
  3. Deposit pheromones on the edges of the graph
     - Quality of the solution
  4. Retain best solution

ACO for shape correspondence

- For a fixed number of iterations:
  1. Traverse the graph
     - Heuristic information and pheromones
  2. Evaluate solution
     - Objective function
  3. Deposit pheromones on the edges of the graph
     - Quality of the solution
  4. Retain best solution

Cost function: QAP formulation

$$\text{QAP} = \qquad \text{Descriptor distance} \quad + \quad \text{Proximity}$$

Cost function: QAP formulation

$$\text{QAP} = (1 - \nu)\,\text{Descriptor distance} \quad + \quad \nu\,\text{Proximity}$$

Cost function: QAP formulation

$$\text{QAP} = (1 - \nu)\,\text{Descriptor distance} \quad + \quad \nu\,\text{Proximity}$$

$$\text{Descriptor distance} = \sum_{i \in I} \text{dist}(R_i, R_{\pi(i)})$$

$$\text{QAP} = (1-\nu)\,\text{Descriptor distance} \quad + \quad \nu\,\text{Proximity}$$

$$\text{Descriptor distance} = \sum_{i \in I} \exp\left(-\frac{\text{dist}(R_i, R_{\pi(i)})^2}{\sigma_R^2}\right)$$

$$\text{QAP} = (1-\nu)\,\text{Descriptor distance} \quad + \quad \nu\,\text{Proximity}$$

$$\text{Descriptor distance} = 1 - \frac{1}{|I|} \cdot \sum_{i \in I} \exp\left(-\frac{\text{dist}(R_i, R_{\pi(i)})^2}{\sigma_R^2}\right)$$

Cost function: QAP formulation

$$\text{QAP} = (1-\nu)\,\text{Descriptor distance} \quad + \quad \nu\,\text{Proximity}$$

$$\text{Descriptor distance} = 1 - \frac{1}{|I|} \cdot \sum_{i \in I} \exp\left(-\frac{\text{dist}(R_i, R_{\pi(i)})^2}{\sigma_R^2}\right)$$

$$\text{Proximity} = \sum_{i \in I} \sum_{i' \neq i \in I} \left|\text{dist}(i, i') - \text{dist}(\pi(i), \pi(i'))\right|$$

Cost function: QAP formulation

$$\text{QAP} = (1-\nu)\,\text{Descriptor distance} \quad + \quad \nu\,\text{Proximity}$$

$$\text{Descriptor distance} = 1 - \frac{1}{|I|} \cdot \sum_{i \in I} \exp\left(-\frac{\text{dist}(R_i, R_{\pi(i)})^2}{\sigma_R^2}\right)$$

$$\text{Proximity} = \sum_{i \in I} \sum_{i' \neq i \in I} \exp\left(\frac{-\text{dist}(i,i')^2}{\sigma_I^2}\right) \left|\text{dist}(i,i') - \text{dist}(\pi(i), \pi(i'))\right|$$

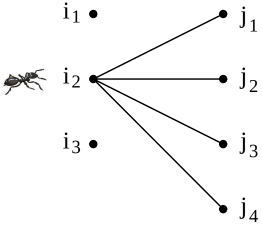
Cost function: QAP formulation

$$\text{QAP} = (1-\nu)\,\text{Descriptor distance} \quad + \quad \nu\,\text{Proximity}$$

$$\text{Descriptor distance} = 1 - \frac{1}{|I|} \cdot \sum_{i \in I} \exp\left(-\frac{\text{dist}(R_i, R_{\pi(i)})^2}{\sigma_R^2}\right)$$

$$\text{Proximity} = \frac{\sum_{i \in I} \sum_{i' \neq i \in I} \exp\left(\frac{-\text{dist}(i,i')^2}{\sigma_I^2}\right) \left|\text{dist}(i,i') - \text{dist}(\pi(i), \pi(i'))\right|}{|I|(|I|-1)/2}$$

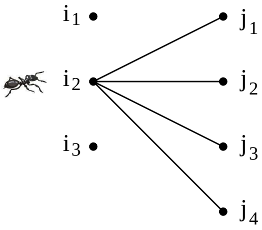
Edge traversal



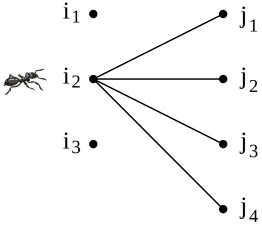
Edge traversal

Edge probability:

$$p_{ij} = \alpha \, \text{Pheromones} + (1 - \alpha) \, \text{Heuristic}$$



Edge traversal



Edge probability:

$$p_{ij} = \alpha \, \text{Pheromones} + (1-\alpha) \, \text{Heuristic}$$

Heuristic information:

$$\text{Heuristic}^{-1} = \text{dist}(R_i, R_j) \times$$

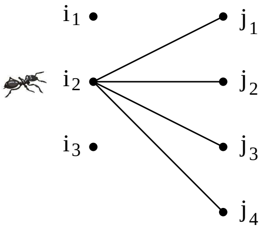
Descriptor similarity

$$|\text{dist}(i, i') - \text{dist}(j, \pi(i'))| \times$$

Proximity to last vertex

$$|\text{dist}(i, i'') - \text{dist}(j, \pi(i''))| \times$$

Proximity to second last vertex



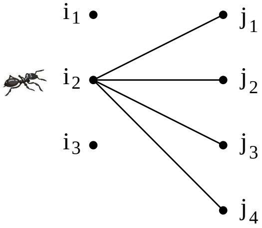
Edge probability:

$$p_{ij} = \alpha \, \text{Pheromones} + (1 - \alpha) \, \text{Heuristic}$$

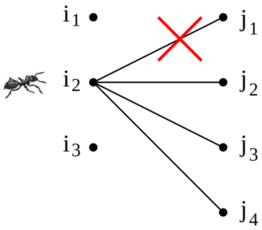
Heuristic information:

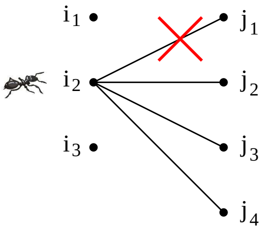
$$\text{Heuristic} = \left(e^{\frac{-\text{dist}(R_i, R_j)^2}{\sigma_R^2}} \right) \times$$
$$\left(1 - e^{\frac{-\text{dist}(i,i')^2}{\sigma_I^2}} \left| \text{dist}(i, i') - \text{dist}(j, \pi(i')) \right| \right) \times$$
$$\left(1 - e^{\frac{-\text{dist}(i,i'')^2}{\sigma_I^2}} \left| \text{dist}(i, i'') - \text{dist}(j, \pi(i'')) \right| \right)$$

Order preservation



- Hard constraints: by removing edges that should not be traversed
    - Order preservation





- Hard constraints: by removing edges that should not be traversed
  - Order preservation
- Soft constraints: by modifying the probabilities

List of ACO parameters and values

ACO Parameters	Symbol	Value
Number of ants	m	1
Number of iterations	T	1000
Influence of pheromones	α	0.3
Pheromone evaporation rate	ρ	0.1
Pheromone deposition constant	δ	0.01
Initial pheromone levels	τ_0	1
Minimum pheromone levels	$\tau_{\min}$	$0.1 \cdot \frac{1}{\lvert I \rvert}$
Influence of proximity	ν	0.7
Gaussian width in $\mathscr{X}$	σ_I	$0.1 \cdot I_{\max}$
Gaussian width in $\mathscr{S}$	σ_R	$0.1 \cdot R_{\max}$

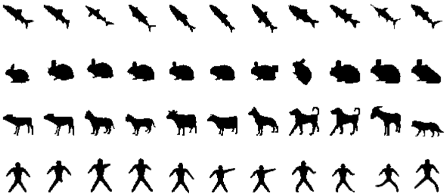
Parameters used in our ACO algorithm and their chosen values

The values are fixed in all the experiments

Outline

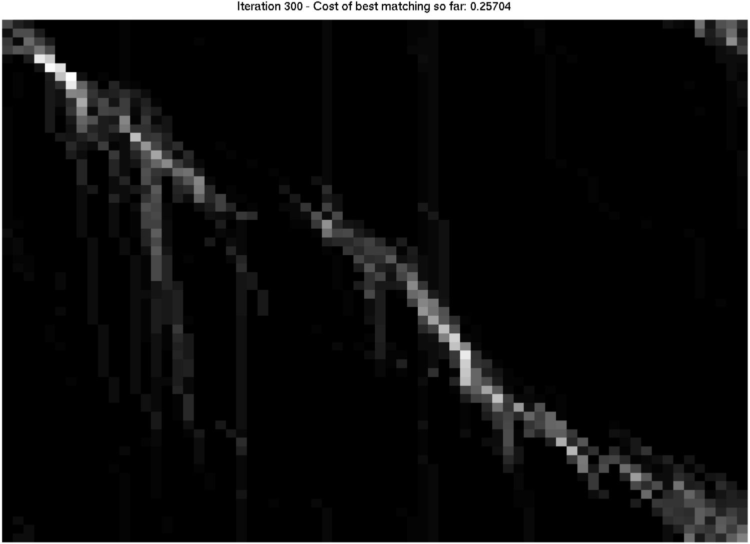
Experimental results

- Contours extracted from the Brown dataset (Sharvit et al., 1998)

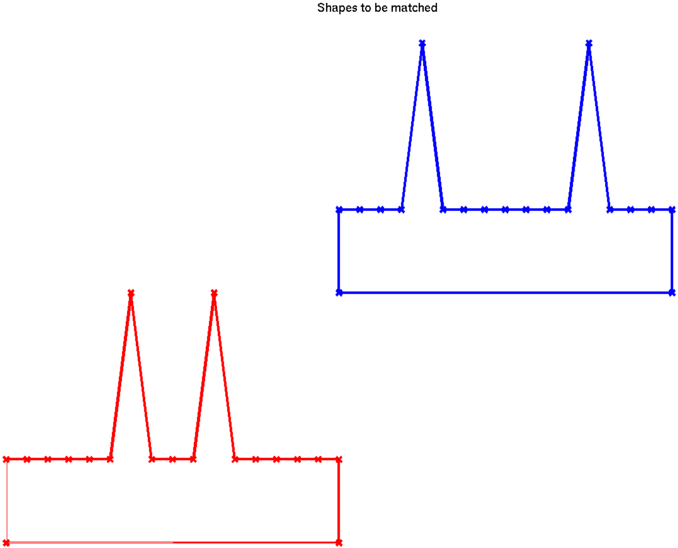


- Shape context (rotation-variant) descriptor (Belongie et al., 2002)

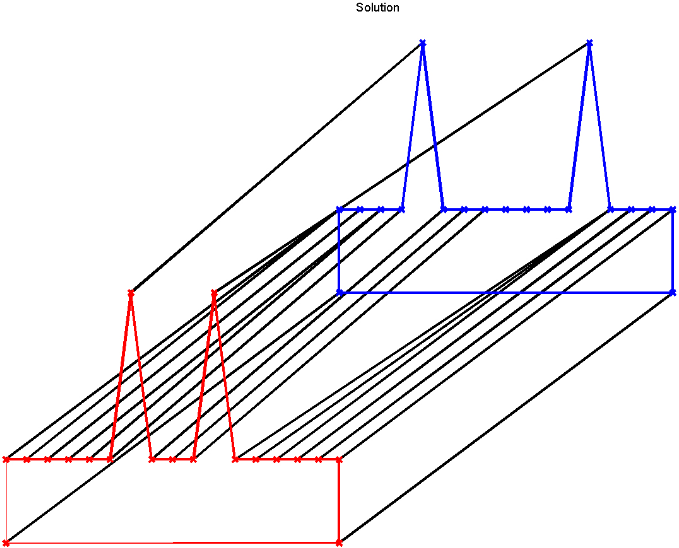
ACO pheromone deposition



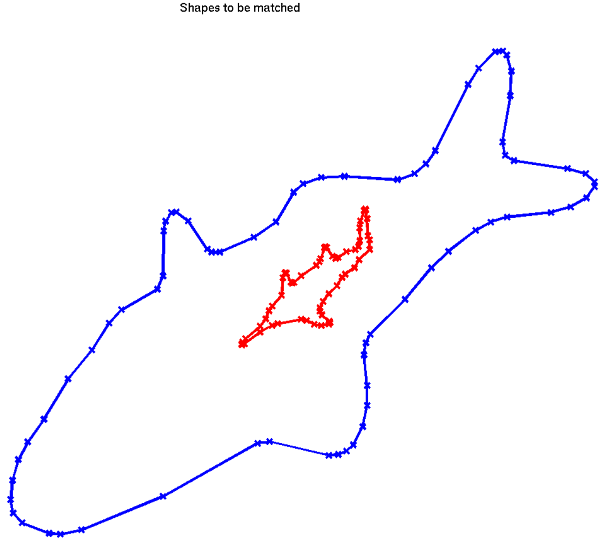
ACO solution computation



ACO solution computation

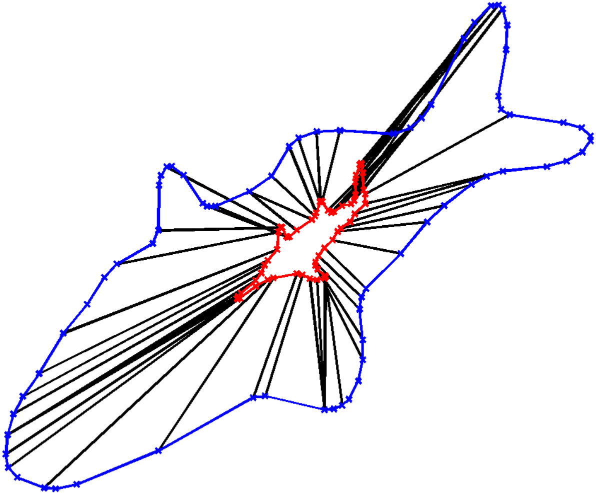


ACO solution computation

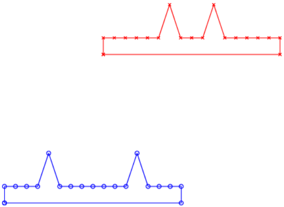


ACO solution computation

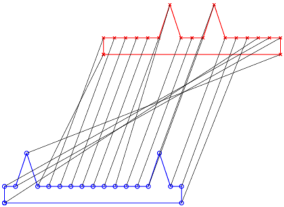
Solution



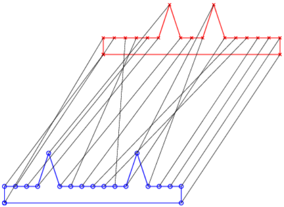
Order preservation (OP) vs. proximity



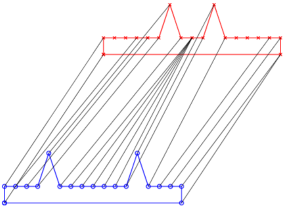
(a) Contours to match



(b) OP

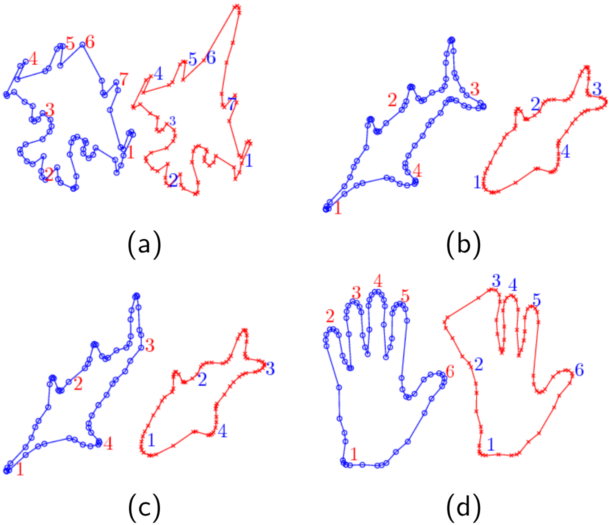


(c) ACO (Proximity only)

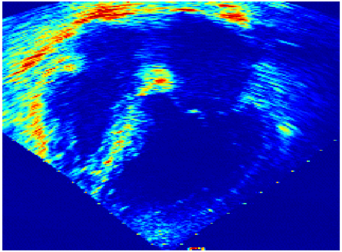


(d) ACO (Proximity + OP)

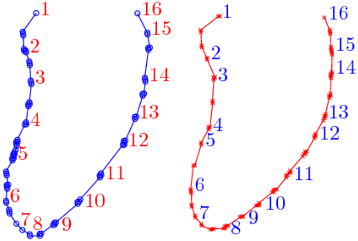
Handling of occlusion or missing parts



Matchings computed by ACO for contours
with occlusion or structure change



(a) Frame



(b) Matching

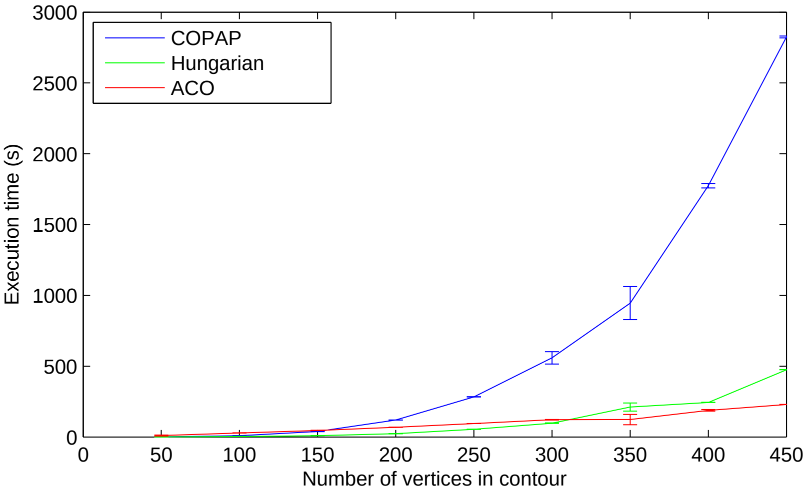
Matching computed by ACO for an open contour of a left ventricle

Evaluation against ground-truth correspondence

- Ground truth provided by a human user
- Error is the sum of geodesic distances between corresponded vertices and ground truth (Karlsson and Ericsson, 2006)
- Compared to Hungarian and COPAP (Scott and Nowak, 2006)

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**

Deviation from ground truth



Execution time comparison between Hungarian, COPAP, and ACO

Outline

Conclusions and future work

- Proximity is incorporated
- The QAP is solved using the proposed ACO algorithm

Conclusions and future work

- Proximity is incorporated
- The QAP is solved using the proposed ACO algorithm

- **Advantages** include
  - Proximity improves the results
  - Results are generally better
  - Resource requirements scale moderately
  - Applicable to contours and unorganized 2D point sets
  - Hard and soft constraints can be easily incorporated

Conclusions and future work

- Proximity is incorporated
- The QAP is solved using the proposed ACO algorithm

- **Advantages** include
  - Proximity improves the results
  - Results are generally better
  - Resource requirements scale moderately
  - Applicable to contours and unorganized 2D point sets
  - Hard and soft constraints can be easily incorporated

- Future work
  - Extension to 2D manifolds
  - Parallelization

Thank you!

Funding for this project was provided by

- Faculty of Applied Sciences
  Simon Fraser University

SFU FAS

MIAL
MEDICAL IMAGE ANALYSIS LAB

gruvi graphics + usability + visualization