Using Genetic Algorithms for Model-Based Object Recognition

George Bebis, Sushil Louis and Yaakov Varol

Department of Computer Science
University of Nevada
Reno NV 89557
bebis@cs.unr.edu

CISST'98

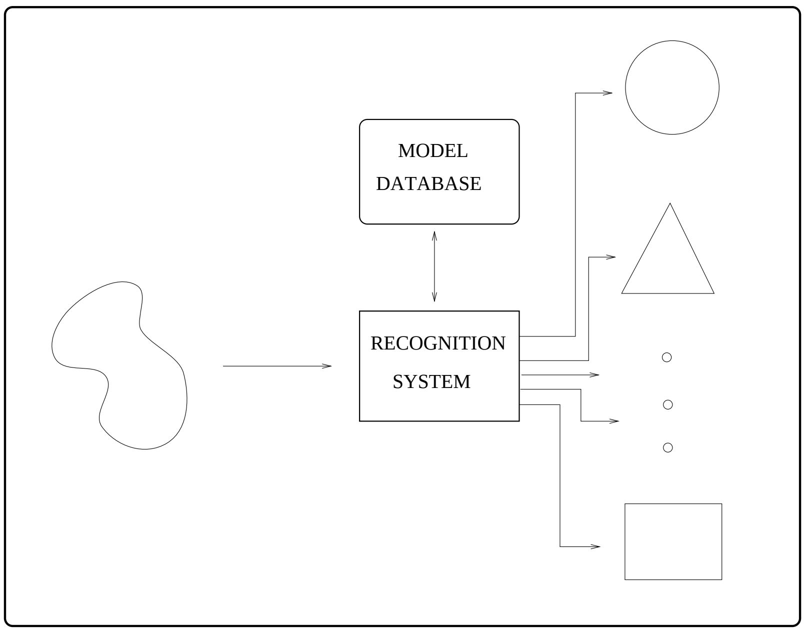
July 6-9, 1998

Las Vegas, Nevada

MODEL-BASED OBJECT RECOGNITION

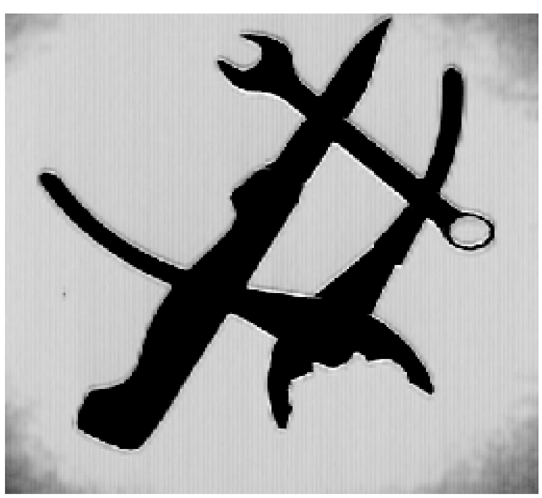
- **Overview**

  - Environment is rather constrained.
  - Search is confined within a finite set of observable models.



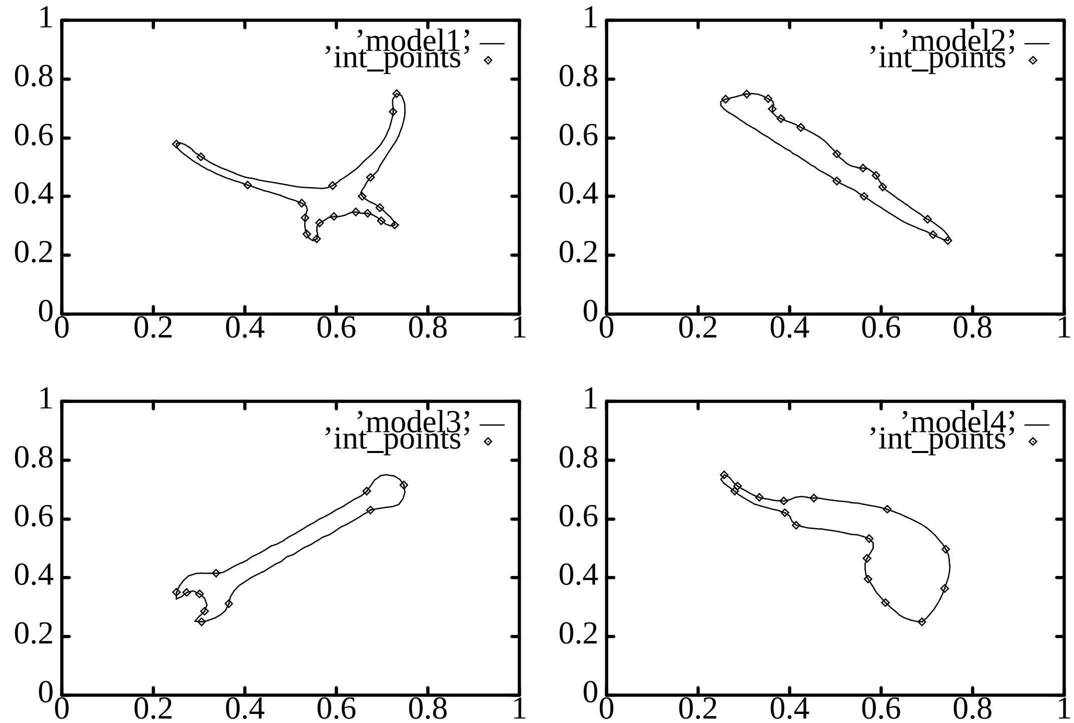
- **Recognition requirements**

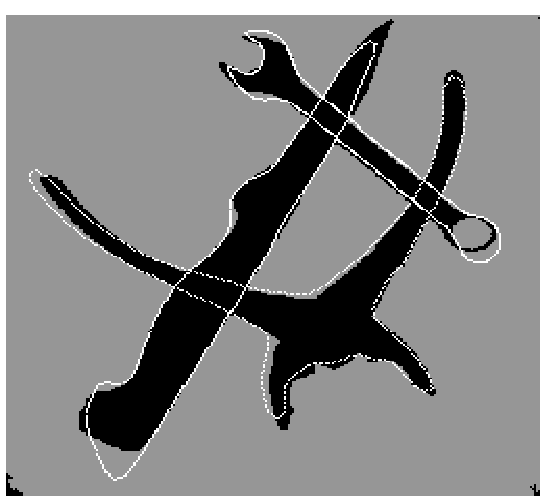
  - Invariant to translation, rotation, and scale.
  - Robust to noise and occlusion.



- **Goal of recognition**

  - The recovery of a geometric transformation which aligns the model(s) with the scene.





• **Planar Objects and 2D Affine Transformations**

- Assume "weak perspective" projection.

- Different views of the same planar object are related through an *affine transformation*.

$$p' = Ap + b$$

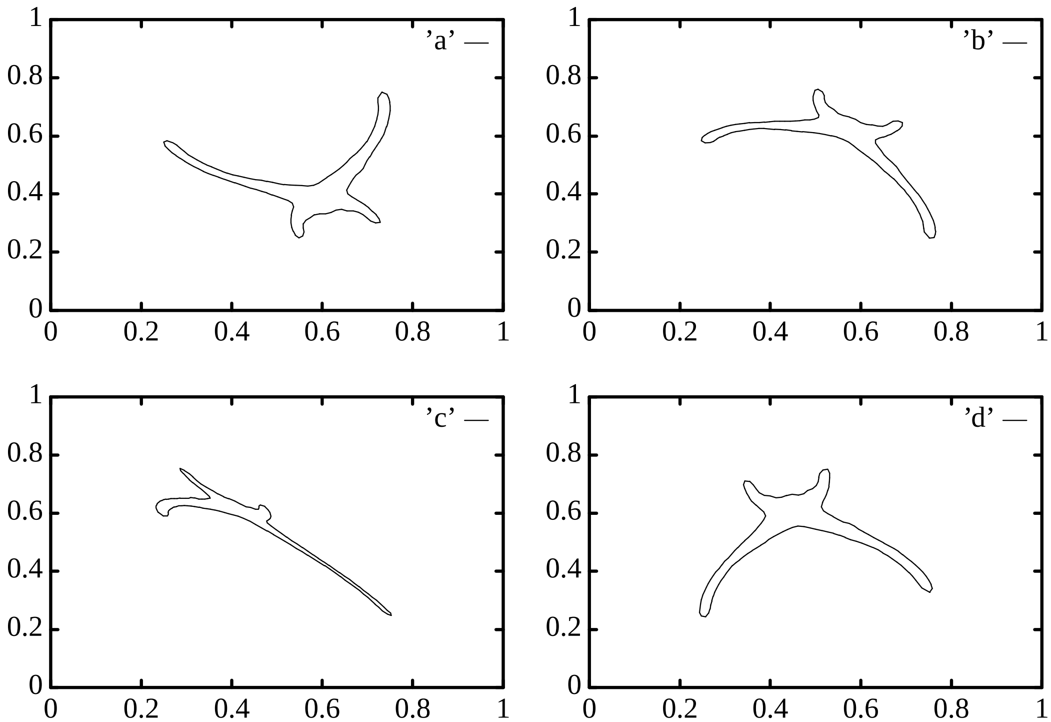


IMAGE-SPACE APPROACHES

- **Procedure**

  - Identify a set of features from the unknown scene which approximately match a set of features from a model object.

  - Recover the geometric transformation that the model object has undergone.

- **Examples of methods in this category**

  - Interpretation tree (Grimson & Lozano-Perez, 1987)

  - Alignment (Huttenlocher and Ullman, 1990)

  - Geometric hashing (Lamdan et al., 1990)

TRANSFORMATION-SPACE APPROACHES

- **Procedure**

  - Search the space of possible transformations.

  - Find a transformation which aligns a large number of model features with the scene.

- **Examples of methods in this category**

  - Hough-transform based methods (Ballard, 1981).

  - Pose clustering techniques (Cass, 1988)

GENETIC ALGORITHMS (GAs)

- **Overview**

  - Parallel search algorithms based on the mechanics of natural selection.

  - Operate iteratively on a population of structures.

  - Each structure represents a candidate solution.

  - Structures are modified at each iteration using selection, crossover, and mutation.

- **Why using GAs for Object Recognition ?**

  - Genetic algorithms were designed to efficiently search large solution spaces.

  - Both the image and transformation spaces are very large !!

  - *Image space*: $O(M^3 S^3)$ possible alignments.

  - *Transformation space*: much larger !! (six dimensional)

- **Previous use of GAs in Image Processing/Analysis**
  - Feature selection (Roth and Levive, 1994)
  - Image segmentation (Swets and Punch, 1995)
  - Target recognition (Katz and Thrift, 1994)
  - Object recognition (Singh et al., 1997, Ansari et al., 1992)
  - Image registration (Fitzpatrick et al., 1984)

- **Problem and Approaches**
  - Recognize real, planar, objects from 2D images assuming that the viewpoint is arbitrary.
  - Genetic search in the image space (GA-IS)
  - Genetic search in the transformation space (GA-TS)

- **Important issues**
  - How to encode solutions ?
  - How to modify solutions ?
  - How to evaluate solutions ?

IMAGE-SPACE GENETIC SEARCH

- **Encoding**

  - At least three model-scene point matches are need to compute the affine transformation.

  - Chromosome contains the binary encoded identities of the three pairs of points.

  - Model points: 19 (5 bits)

  - Scene points: 19 - 45 (6 bits)

  - Chromosome length: 3 x 5 + 3 x 6 = 33 bits

Model pt 1	**Model pt 2**	**Model pt 3**	**Scene pt 1**	**Scene pt 2**	**Scene pt 3**

<— **5 bits** —> <— **6 bits** —>

- **Fitness evaluation**

  1. Compute affine transformation.
  2. Apply the transformation on all the model points.
  3. Compute the error (BE) between transformed model points and scene points.

$$BE = \sum_{i=1}^{M} d_j^2$$

(d_j min distance between the j-th model point and the scene)

$$Fitness = 10000 - BE$$

ESTIMATING THE RANGES OF VALUES OF THE PARAMETERS OF AFFINE TRANSFORMATION

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \cdots & \cdots & \cdots \\ x_M & y_M & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ b_1 \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ \cdots \\ x'_M \end{bmatrix} \quad or \quad Pc_1 = p_{x'} \qquad (1)$$

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \cdots & \cdots & \cdots \\ x_M & y_M & 1 \end{bmatrix} \begin{bmatrix} a_{21} \\ a_{22} \\ b_2 \end{bmatrix} = \begin{bmatrix} y'_1 \\ y'_2 \\ \cdots \\ y'_M \end{bmatrix} \quad or \quad Pc_1 = p_{y'} \qquad (2)$$

- Assume that the image coordinates of the unknown views $(p_{x'}, p_{y'})$ are restricted to belong to a given interval, (e.g., by scaling the image coordinates in [0,1]).

- Use **Interval Arithmetic** to find all the possible solutions of (1) and (2) assuming that $p_{x'}$ and $p_{y'} \in [0,1]$.

$$Pc_1^I = p_{x'}^I$$

$$Pc_2^I = p_{y'}^I$$

- Solving (1) and (2) using **Singular Value Decomposition**

$$P = U_P W_P V_P^T$$

$$c_1 = P^+ p_{x'} \quad or \quad c_1 = \sum_{i=1}^{3} (\frac{u_i p_{x'}}{w_{ii}}) v_i \tag{3}$$

$$c_2 = P^+ p_{y'} \quad or \quad c_2 = \sum_{i=1}^{3} (\frac{u_i p_{y'}}{w_{ii}}) v_i \tag{4}$$

Evaluate (3) and (4) using Interval Arithmetic (Moore, 1966)

$$t = [t_1, t_2], r = [r_1, r_2]$$

$$t + r = [t_1 + r_1, t_2 + r_2]$$

$$t * r = [min(t_1 r_1, t_1 r_2, t_2 r_1, t_2 r_2), max(t_1 r_1, t_1 r_2, t_2 r_1, t_2 r_2)]$$

- Apply *preconditioning* to optimize the ranges.

The computed ranges of values.

Ranges of values			
	range of a11	range of a12	range of b1
original	[-2.953, 2.953]	[-2.89, 2.89]	[-1.662, 2.662]
preconditioned	[-0.408, 0.408]	[-0.391, 0.391]	[0.0, 1.0]

TRANSFORMATION-SPACE GENETIC SEARCH

- **Encoding**
  - Each chromosome contains six fields.
  - Only the range of each coefficient needs to be represented.

    a_{11} assumes values in [-0.408, 0.408]

    Its range is: 0.408 - (-0.408) = 0.816

    2 decimal digit accuracy: 82 values must be encoded.

    7 bits are needed to encode 82 values.

- **Decoding**
  - Some encoded solutions might be invalid.

    7 bits can encode at most 128 values.

    [0, 127] should be mapped to [0, 81]

    $a_{11} = MIN(a_{11}) + (82/2^7)) * Decimal(W)$

    (W is the binary encoded solution corresponding to a_{11})

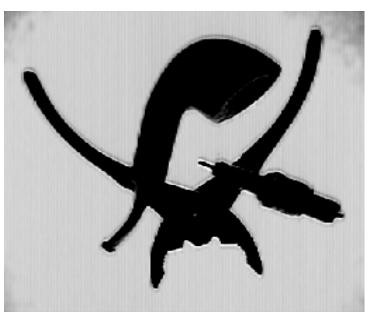
- **Fitness evaluation**
  - Same as before (less costly to compute now)

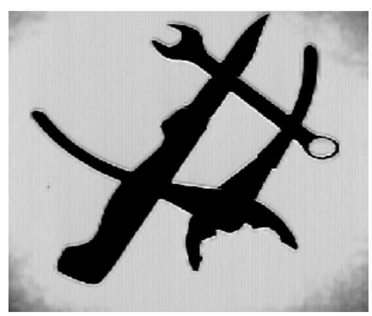
<u>GENETIC OPERATORS</u>

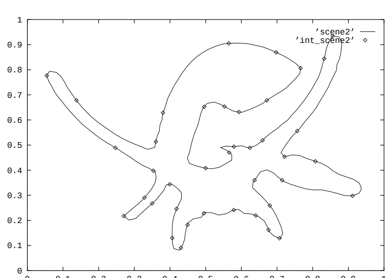
- Two-point crossover (*prcoss*: 0.95).
- Point mutation (*pmut*: 0.05).
- Cross generational selection strategy.
- Fitness scaling (*scaling factor*: 1.2).

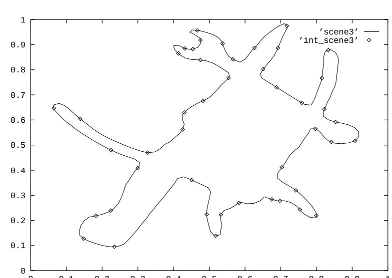
<u>SIMULATIONS AND RESULTS</u>

- Three scenes (S1, S2, S3) of increasing complexity.
- S2, S3 are shown below (S1 was the same as model).
- 10 trials per scene.









- **Parameters**

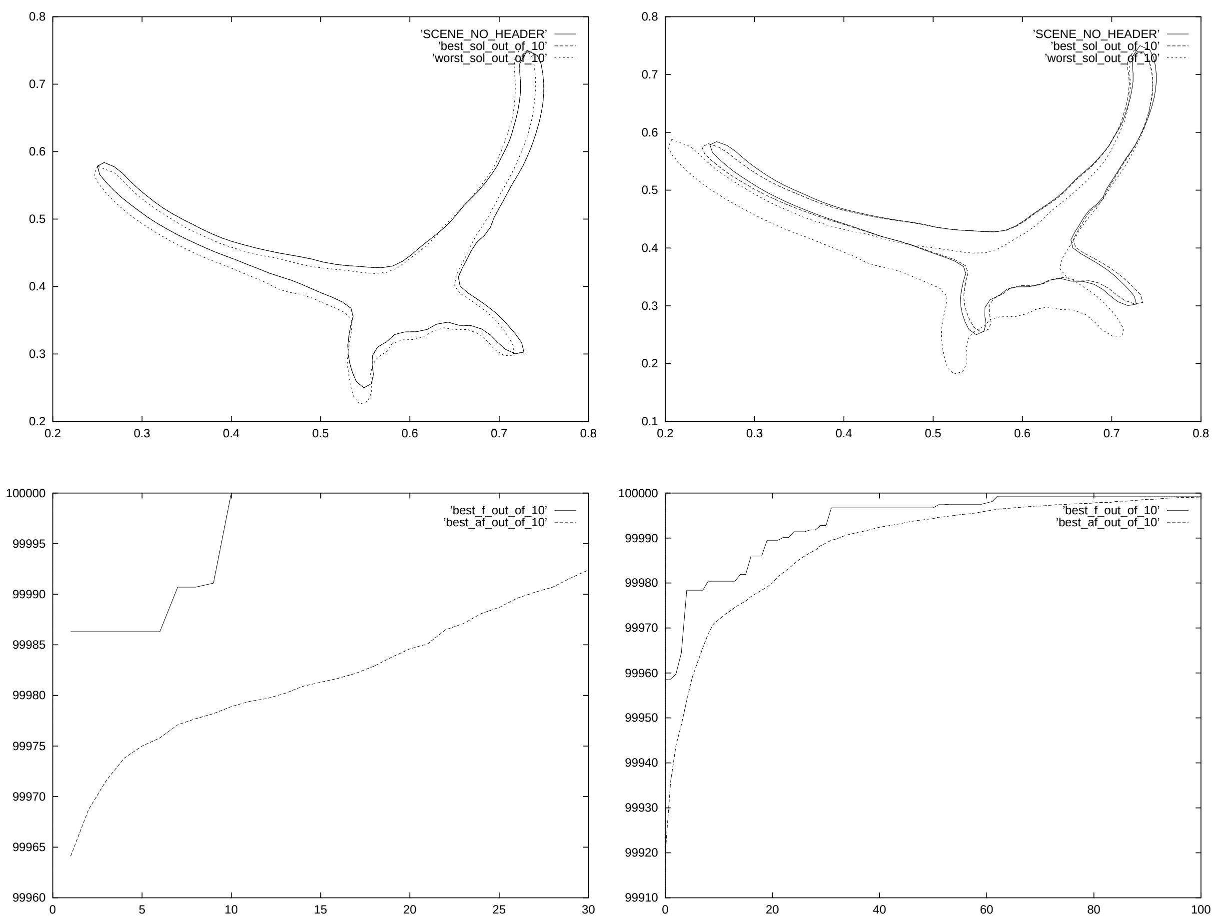
Values of Parameters			
	S1	S2	S3
Population Size	100	200	500
Generations (GA-IS)	30	50	50
Generations (GA-TS)	100	100	100

  - All other parameters were the same for all scenes.

- **Scene1**

  - Correct solutions were found in all 10 trials.

GS-IS GA-TS

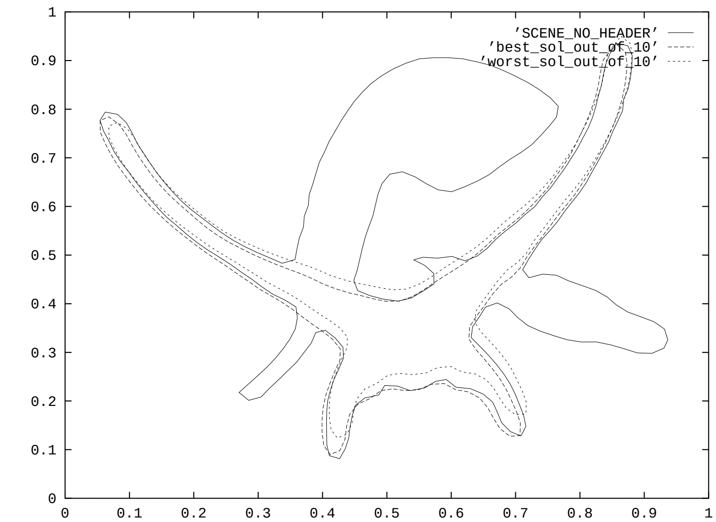


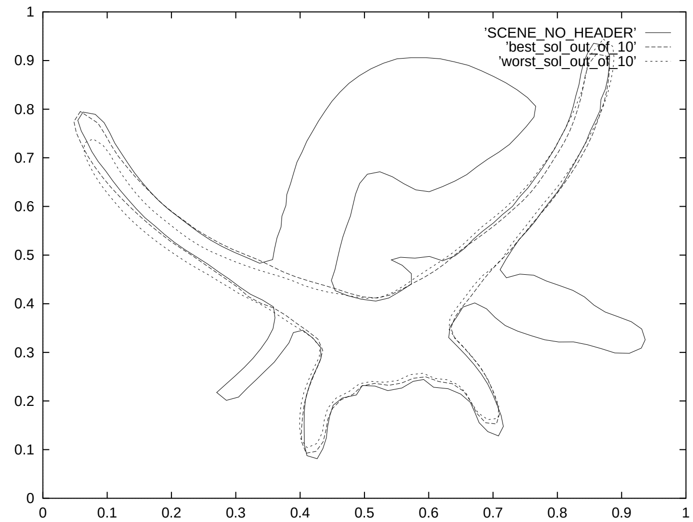
- **Scene2**

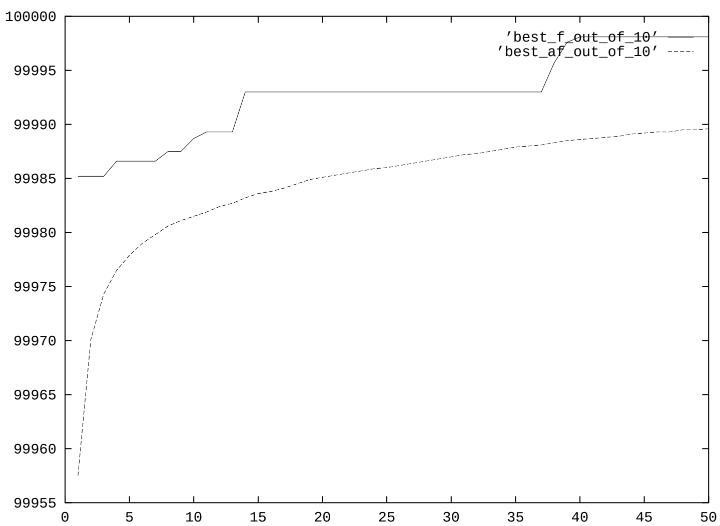
  - Correct solutions were found in all 10 trials.

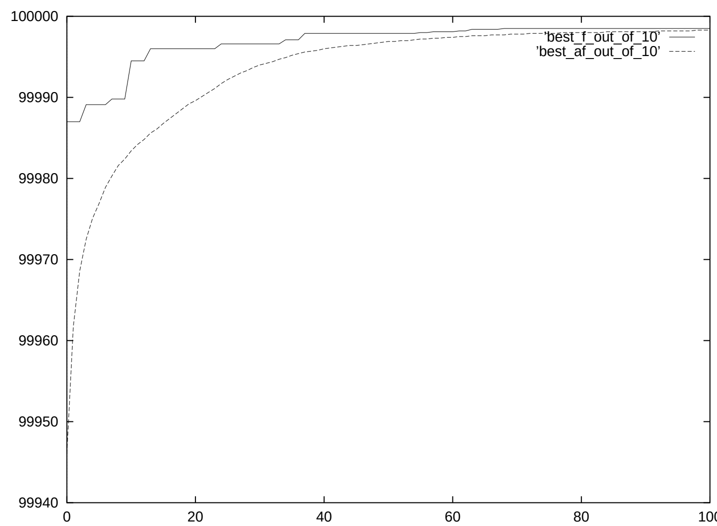
GS-IS

GA-TS





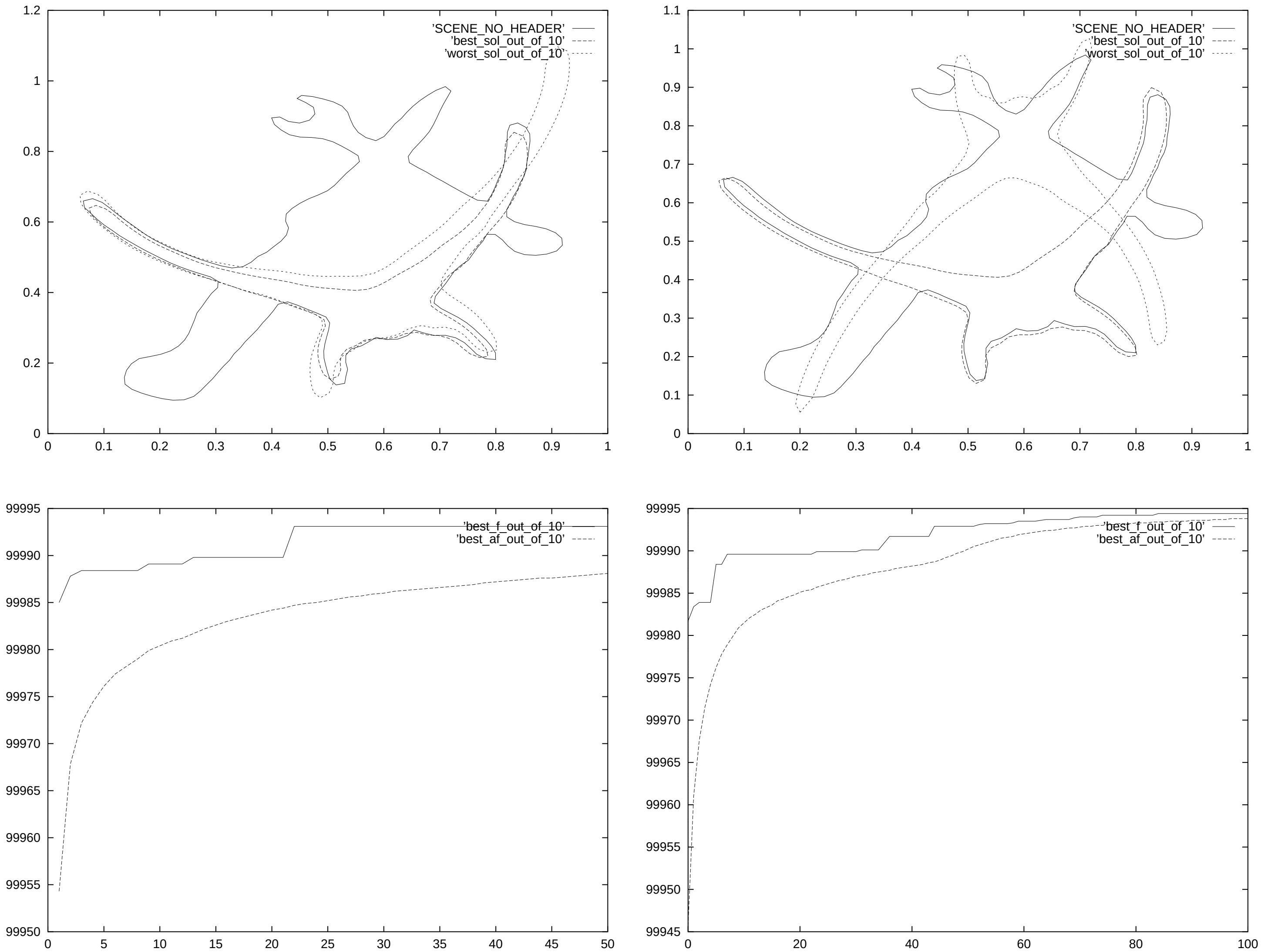




- **Scene3**

  - The GA-TS approach missed the correct solution once.

GS-IS GA-TS



$$M_3 = \binom{19}{3} = 969$$

$$Total\ \ number\ \ of\ \ matches = 3!\ \ x\ \ M_3\ \ x\ \ S_3$$

Summary of results (GA-IS approach).

Results			
Scene	Scene Points	Number of Matches	$GA - IS_{matches}$
Scene1	19	5,633,766	1800(0.0003)
Scene2	40	57,442,320	47,800(0.0008)
Scene3	45	82,500,660	133,250(0.0016)

$$Total\ \ number\ \ of\ \ possible\ \ transformations:$$

$$82^2 x 79^2 x 101^2 = 428,079,701,284$$

Summary of results (GA-TS approach).

Results		
Scene	Number of Transforms	$GA - TS_{matches}$
Scene1	428,079,701,284	8010 (0.000000018)
Scene2	428,079,701,284	8760(0.00000002)
Scene3	428,079,701,284	8620(0.00000002)

• **Conclusions**

- Exact and near exact matches were found reliably and quickly.

- GA-TS converges faster.

- GA-IS finds better solutions.

- GAs are a viable tool for searching the image and transformation spaces efficiently.

• **Future work**

- Incorporate constraints into the fitness function (e.g, geometric constraints).

- Consider more than one models.

- Extend the work to the case of real 3D objects.

- Consider parallel implementations for real time performance.