

A Modified Genetic Algorithm for Matching Building Sets with the Histograms of Forces

Andrew R. Buck, James M. Keller, *Fellow IEEE*, Marjorie Skubic, *Member IEEE*

Abstract—This paper presents an approach to the task of locating a group of buildings based solely on their relative spatial relationships. This situation can occur in the problem of conflation of a hand or machine drafted map to a satellite image or in matching of two images taken under different viewing conditions (the correspondence problem). Of importance to us is the general text-to-sketch problem where a sketch of building locations must be matched to actual satellite imagery. Information about the nature of these relative positions is captured by the histograms of forces. In this paper, we consider a modified genetic algorithm that allows us to search for a specific group of buildings within a large geospatial database using the histograms of forces in the matching process. A novel mutation operator is introduced to adapt the standard GA to this environment.

I. INTRODUCTION

IN the general problem of text-to-sketch (T_2S), a linguistic description of a location is converted into a pictorial representation [1]. A person who is in an unfamiliar place may describe his location using nearby landmarks and their relative positions to one another. From these linguistic descriptions, the T_2S system in [1] can create an artificial image which represents the relative locations and sizes of the recognized landmarks. This sketch can assist in finding the real-world location of the person. For example, Figure 1 shows a sketch generated by the T_2S system and the corresponding location in a satellite image. This sketch was created from a series of linguistic descriptions that take a form similar to: “I see a long rectangular building on my left and a small L-shaped building on my right.”

The sketch generated either by the T_2S system or by hand is rarely a completely accurate representation of the actual environment. Inaccuracies in the sketch and in the segmentation of the satellite image make comparing the two a challenging task. A system that can match the generated sketch to segmented satellite imagery must be tolerant of mistakes in shape and scale, yet sensitive to relative relationships between objects. The most accurate feature of a sketch tends to be the spatial relationships of the objects within, rather than their specific shapes and sizes. We use the histograms of forces (HoF) [2] as a means of capturing the relative orientation and scale between two objects. When a histogram is created between each pair of objects in a sketch, it provides

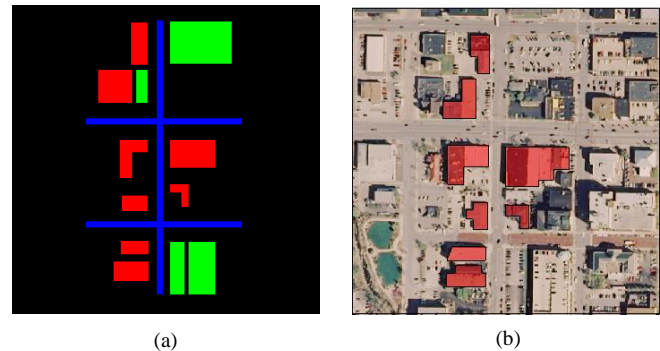


Fig. 1 An example of the building matching problem. (a) A general sketch of building shapes and locations shown in red, with additional information such as parking lots (green) and roads (blue). (b) A real-world location with buildings within a large image database that closely match the sketch in (a).

a quantitative metric which can be used to match it to a real location. The similarity of the HoF to a natural human description, along with its use of relative rather than absolute relationships, makes it an ideal candidate for comparison of object configurations.

To solve the problem of comparing the objects in a sketch to those in a segmented satellite image, we consider the use of a modified genetic algorithm. Genetic algorithms (GAs) are often used for searches over a large search space where an exhaustive search would be impractical [3]. They are a branch of evolutionary computation which seeks to find a suitable solution by creating a population of random individuals and evolving them into better solutions. The classical GA developed by Holland [4] operated on chromosomes represented by bitstrings and iteratively applied selection with simple mutation and crossover operations. Later research adapted the classical GA to a variety of problem domains, including the problem of performing a spatial search. GAs have been used for matching 2D line segments by Beveridge [5], who explored the benefits of a messy genetic algorithm and by Rodríguez [6], who used a topological framework to search spatial configurations. In this paper, we design a chromosome representation and mutation operator to take advantage of the spatial knowledge captured by the histograms of forces to match a set of nearby buildings in a confined neighborhood.

We begin by reviewing the theory behind the histograms of forces and genetic algorithms. The HoF GA modification to the classical GA used in this paper is then explained in detail. We conclude the paper with an analysis of the performance of the HoF GA on a large database of buildings. We consider matching sketches both with and without global

Manuscript received May 1, 2010. This work is funded by the U.S. National Geospatial-Intelligence Agency NURI grant HM 1582-08-1-0020.

A. R. Buck, J. M. Keller, and M. Skubic are with the Department of Electrical and Computer Engineering, University of Missouri-Columbia, MO 65211, USA (e-mail: arb9p4@mail.mizzou.edu; kellerj@missouri.edu; skubicm@missouri.edu).

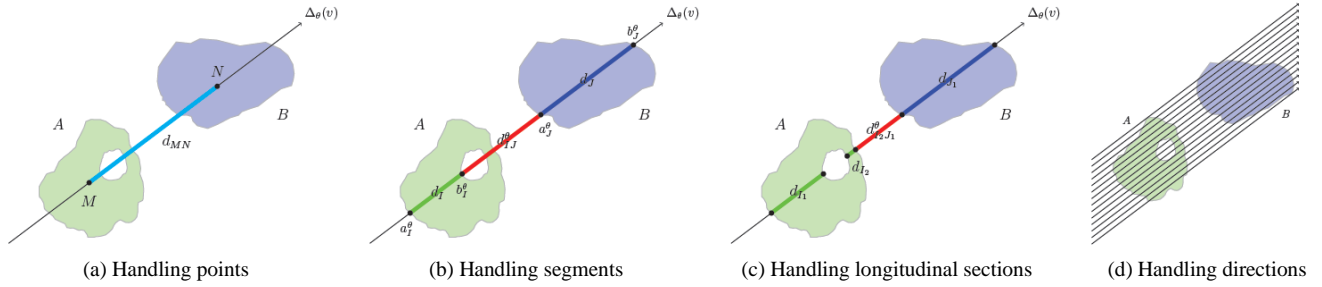


Fig. 2 The computation of the histograms of forces for two crisp objects, A and B for a single value θ . Functions are defined which evaluate (a) points: $\phi_r(M - N) = \frac{1}{d_{MN}^r}$, (b) line segments: $f_r(d_I, d_{IJ}^\theta, d_J) = \int_{a_I^\theta}^{b_I^\theta} \int_{a_J^\theta}^{b_J^\theta} \phi_r(u - v) dv du$, (c) longitudinal sections: $(\theta, A_\theta(v), B_\theta(v)) = \sum_{i,j} f_r(d_{I_i}, d_{IJ}^\theta, d_{J_j})$, and (d) directions: $F_r^{AB}(\theta) = \int_{-\infty}^{\infty} \mathcal{F}_r(\theta, A_\theta(v), B_\theta(v)) dv$.

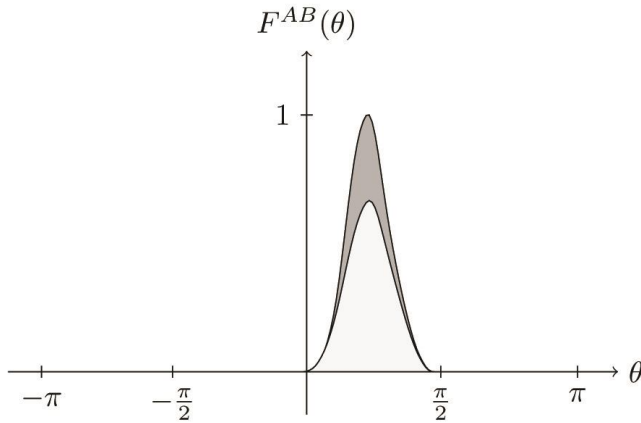


Fig. 3 The histograms of constant (dark gray) and gravitational (light gray) force from object A to B in Figure 2. The concentration of values in the range 0 to $\frac{\pi}{2}$ implies that B is above and to the right of A .

transformations, and include an example of matching a sketch from the T_2S system.

II. BACKGROUND

A. Histograms of Forces

Given a pair of crisp, two-dimensional objects A and B , and a direction θ , we want to know the degree to which A is in direction θ from B . We define a function $F_r^{AB}(\theta)$ that evaluates the amount of support for which this is true. To construct $F_r^{AB}(\theta)$, we must evaluate the objects A and B first as sets of points, then line segments, longitudinal sections, and finally general directions [7]. Given a line $\Delta_\theta(v)$ as in Figure 2, we can determine the set of line segments that intersect both objects A and B . For each point M in object A , we measure the distance d_{MN} to a point N in object B . We can then define a function that evaluates the distance between two points,

$$\phi_r(M - N) = \frac{1}{d_{MN}^r}. \quad (1)$$

For each pair of line segments I and J , which are contained within and bounded by A and B respectively, we define a function which integrates the distances between all possible pairs of points in I and J ,

$$f_r(d_I, d_{IJ}^\theta, d_J) = \int_{a_I^\theta}^{b_I^\theta} \int_{a_J^\theta}^{b_J^\theta} \phi_r(u - v) dv du. \quad (2)$$

The summation of this function over all possible line segments contained within and bounded by A and B in a direction θ gives the force value \mathcal{F}_r , defined as

$$\mathcal{F}_r(\theta, A_\theta(v), B_\theta(v)) = \sum_{i,j} f_r(d_{I_i}, d_{IJ}^\theta, d_{J_j}). \quad (3)$$

Evaluating \mathcal{F}_r over the set of parallel lines in a direction θ gives the force histogram $F_r^{AB}(\theta)$:

$$F_r^{AB}(\theta) = \int_{-\infty}^{\infty} \mathcal{F}_r(\theta, A_\theta(v), B_\theta(v)) dv. \quad (4)$$

We can evaluate these functions with different values of r to capture different information. For example, $r = 0$ gives constant forces, which represents object independence from distance, and $r = 2$ gives the gravitational force, which represents object independence from scale. Figure 3 shows the histograms of constant and gravitational force between objects A and B from Figure 2. These histograms represent the relative position of the two objects in a way that preserves the direction and relative scale, but does not distinguish between small variations in shape. This is the primary reason for choosing the HoF as the representation scheme for the matching algorithm.

B. Genetic Algorithms

In the domain of evolutionary computation, genetic algorithms model natural evolution by using genotypes. Each candidate solution is expressed as a chromosome containing a set of genes, one for each variable to be optimized. A fitness function evaluates how well each chromosome matches the ideal solution. Evolution occurs through the selection, crossover, and mutation operators. The selection operator uses the fitness values to pick a set of chromosomes to survive through to the next generation. Crossover creates offspring chromosomes by combining genetic material from a set of parents. Mutation randomly alters the genes of a single chromosome to introduce new genetic material.

The classical GA uses all of these operators on a population of chromosomes represented by bitstrings, where each

bit represents a gene. A population of chromosomes is created randomly and represents the original generation. The selection operator is used to select a set of chromosomes for reproduction. This occurs through the mutation and crossover operations, which are applied to the population with probabilities m_{prob} and c_{prob} respectively. After creating new genetic material through these operations, the selection operator selects a new population from the existing set of chromosomes to serve as the next generation. This cycle of evolution continues until some stopping condition is met, typically a maximum number of generations, or an acceptable solution.

III. ALGORITHM

A. Database Representation

In order to match a sketch to a satellite image, we must define the locations of all the buildings in the search area. This is done by creating a polygon outline of each building's footprint. These coordinates are stored as absolute positions in a database of polygon features, represented in the coordinate system of the scene. From this information, we calculate the HoF between any two buildings. As the size of the search area increases, the number of possible building pairs grows at a rate of $n!$. To limit our search space, we only calculate the HoF for the K nearest neighbors of each building. We can determine the set of nearest neighbors for each building by calculating the set of polygons with the shortest Euclidian distance between any two points in both polygons. The value chosen for K depends on the number and separation of the buildings that will be matched from sketches. This value can have a significant impact on the performance of the algorithm. If the number of nearest neighbors is too small, it may become impossible to find the set of buildings that most closely matches an input sketch. Conversely, if there are too many nearest neighbors, the time required to compute each new generation becomes a limitation.

Our method defines two sets of buildings: the database set containing all the buildings in a segmented satellite image, and the target set containing all the buildings in the sketch to be matched. The database set can be pre-calculated for each search area, but the target set will be unique for each input sketch. Figure 4 shows the database set used in our experiments. Each set assigns an index number to every building and creates a list of each building's nearest neighbors. The HoF are computed for each set and are used by the GA to perform the matching. No absolute position information is included in the HoF, only relative information. This means that matching can be done on a database and target set that use independent coordinate systems. When a match has been found, the buildings corresponding to the matched histograms can be determined by their index numbers.

B. Chromosome Representation

As described by Algorithm I, the runtime parameters for the GA are the polygon outlines of the target set T_{poly} , the population size n , the mutation probability p_m , the maximum

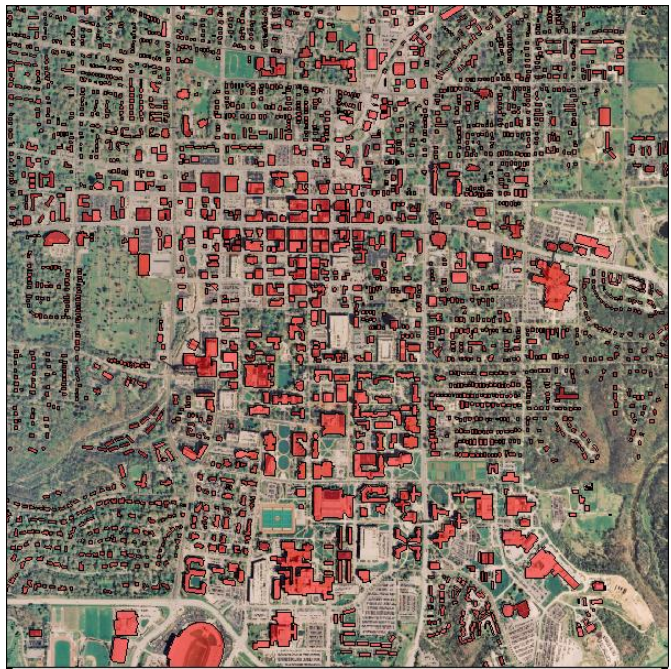


Fig. 4 Database set used for testing the modified genetic algorithm. The set contains 2467 buildings from downtown Columbia, MO and the University of Missouri-Columbia campus.

number of generations t_{max} , and the minimum required fitness value to end the search f_{min} . We first calculate the force histograms between the polygons in the target set and store these as the target histograms T_{HoF} . Every chromosome in the GA is a set of potential building indexes, where each gene represents a building. The force histograms between the buildings of a chromosome are stored as the chromosome's search set S_{HoF} . The goal of the algorithm is to find a candidate solution's search set S_{HoF} that most closely matches the target set T_{HoF} . The algorithm starts by creating a random population of n chromosomes over the search space with the restriction that each set must be fully connected. This implies that there is a histogram from each building in the chromosome to every other building in the chromosome. If the number of nearest neighbors stored for each building in the database set is too low, a complete match may not be possible, and the algorithm will thrash about a suboptimal solution. Ideally the initial chromosomes will be evenly distributed over all the buildings in the database. Each chromosome, then, represents an argument or search scene. Once a chromosome is created, we immediately evaluate its fitness.

The fitness of a chromosome can be found by comparing all of the histograms of its search set S_{HoF} with the histograms of the target set T_{HoF} . Prior to comparison, the search set is rotated to an optimal value, based on the properties of HoFs [2], [7]. This ensures that the best match can be found, even if the target set is rotated with respect to what appears in the search space of the database set. Rotation is performed by finding the centroids of the histograms in the search set $h_S(\theta)$, and the target set $h_T(\theta)$, which can be approximated as the value of θ for which $h(\theta)$ is largest. The centroid of each search histogram is compared to the centroid of each target

Algorithm I: Modified Genetic Algorithm

Input: T_{poly} : target set, n : population size, p_m : mutation probability, t_{max} : max generations, f_{min} : min fitness

Evaluate T_{poly} to produce target HoF set, T_{HoF} ;
 Add n random chromosomes to population, $C(0)$;
 Evaluate the fitness, $f(x(t))$ of each individual;
 Update set of best individuals and f_{best} ;
 $t = 0$;

while ($t < t_{max}$ **and** $f_{best} < f_{min}$) {
 for each chromosome in $C(t)$ {
 Perform mutation with probability p_m {
 Select a gene at random;
 for each nearest neighbor {
 Replace chosen gene with nearest neighbor;
 Evaluate the fitness of the chromosome;
 }
 Replace the chosen gene with the nearest neighbor that has the highest fitness;
 }
 Add either child or parent to $C(t+1)$ based on $P(x_c(t) \text{ wins})$;
 }
 if ($t \bmod 100 == 0$) {
 Add $n/10$ random chromosomes to $C(t)$;
 Select n chromosomes for $C(t+1)$;
 }
 Update set of best individuals and f_{best} ;
 $t = t + 1$;
 }

Output: Set of best individuals and f_{best}

histogram and the average difference is saved as the best angle of the chromosome. Each histogram in the search set is then shifted by the best angle to most closely match the target set. We define N to be the number of histograms in the target set, which must equal the number of histograms in the search set. The rotation process is summarized by equations 5-7.

$$c(h(\theta)) = \operatorname{argmax}_{\theta} h(\theta) \quad (5)$$

$$\alpha = \frac{1}{N} \sum_{\substack{h_S \in S_{HoF} \\ h_T \in T_{HoF}}} c(h_T(\theta)) - c(h_S(\theta)) \quad (6)$$

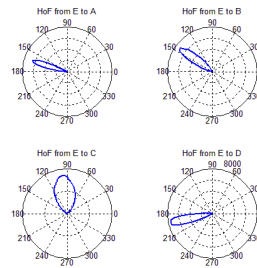
$$h'(\theta) = h(\theta + \alpha), \text{ for all } h(\theta) \in S_{HoF} \quad (7)$$

Histograms are compared using their cross-correlation, which effectively determines the amount that they overlap. This can be calculated using the formula,

$$\mu_c(h_S, h_T) = \frac{\sum_{\theta} h_S(\theta) h_T(\theta)}{\sqrt{\sum_{\theta} h_S^2(\theta) h_T^2(\theta)}} \quad (8)$$



(a)



(b)



(c)

Fig. 5 The process of the novel mutation operator just before convergence (a) and after convergence (c). The chromosome contains buildings A-E (four green and one red). The target set contains buildings A-E (four green and one blue). Building e in red is randomly chosen for replacement. (b) shows the histograms of forces relating E to buildings A-D. The nearest neighbor (shown in yellow except for E in blue) that most closely matches these histograms is chosen to replace building e in red. Because this is the last mutation before convergence, building E is a perfect match. (c) shows the final chromosome, perfectly matched with the target set.

Here, μ_c is in the range $[0, 1]$ and is the cross-correlation between histograms h_S and h_T . The final fitness of the chromosome can be calculated by averaging the cross-correlations of each histogram pair between the search and target sets with the formula,

$$f(x(t)) = \frac{1}{N} \sum_{\substack{h_S \in S_{HoF} \\ h_T \in T_{HoF}}} \mu_c(h_S, h_T). \quad (9)$$

The fitness of a chromosome is in the range $[0, 1]$ where a fitness of zero implies that there is no evidence to support a match between this search set and the target set, and a fitness of one implies a perfect match between the two sets. It should be noted that the ordering of genes in the chromo-

some is important. Two chromosomes with identical genes but in different orders will not be considered the same solution by the fitness function.

C. Mutation Operator

We create new chromosomes for the next generation by applying the mutation operator to the entire population with a probability equal to p_m . Because each chromosome represents a set of buildings that are close together, the combination of sets through crossover could result in child chromosomes that are not fully connected, so mutation is used exclusively. Our mutation operator begins by randomly selecting one of the genes to replace. This gene corresponds to one of the buildings in the target set. The set of potential replacements is the set of buildings which could replace the chosen gene and maintain full connectivity. The fitness of the chromosome is evaluated for each potential replacement. The building that produces the chromosome with the highest fitness value is considered to be the best “fit” and is chosen as the new value for the gene. The mutation operator is further highlighted in Figure 5.

When a chromosome is mutated, its fitness determines if it will replace its parent by the formula,

$$P(x_c(t) \text{ wins}) = \frac{f(x_c(t))}{f(x_c(t)) + f(x_p(t))} \quad (10)$$

where x_c is the child chromosome, x_p is the parent chromosome, and $f(x_i(t))$ is the fitness of chromosome x_i . A parent with a lower fitness value than its child is more likely to be replaced. This helps to prevent a low-fitness child from replacing a high-fitness parent. After all chromosomes have had the opportunity to mutate, the generation counter advances. Every 100 generations, the population size is increased by 10% and a new population is chosen of the original size. This allows new genetic material to be added late in the search process, which keeps chromosomes from getting trapped in local minima. The algorithm continues until a chromosome with the minimum required fitness is found or the generation limit is exceeded.

IV. RESULTS

Our tests were conducted on a dataset of 2467 buildings from Columbia, MO, shown in Figure 4. We created 100 test sets with five buildings each, with the restriction that each building must be one of the 30 nearest neighbors of each other building in the test set. In the first experiment, we evaluate the performance of the HoF GA using direct resubstitution. Next, we apply a random global transformation to each test set and reevaluate the algorithm’s performance. Finally, we look at the matching ability of the HoF GA on an example sketch created by the T₂S system. Tests are performed on a computer running Windows 7 with a Core i7 processor running at 2.93 GHz and 12 GB of RAM.

A. Resubstitution

In our first experiment, we use 100 test sets with five

buildings each and evaluate the average performance of the HoF GA over 10 runs. The database for these trials uses 30 nearest neighbors. We use population sizes of 50, 100, 250, 500, and 1000 with a maximum of 10,000 generations. The mutation probability is fixed at 1 and a fitness value of 0.99 is required for convergence. This accounts for rounding errors that may make it impossible to achieve a perfect fitness value of 1.

To evaluate the performance of the algorithm, we first compute the average top fitness value per generation, as shown in Figure 6a. This is done by averaging the top fitness values within each population for each generation over all tests for a given population size. Our results show that the average top fitness value within the population rapidly increases during the first 50 generations and then begins to level out. By recording the state of the population at each generation, we can generate animations that show the progress of the matching algorithm and the fitness of each chromosome. During these first few generations, the chromosomes select buildings that are in the same general spatial configuration as those in the input sketch. The remaining time that the algorithm runs is spent optimizing the selection of individual buildings to more closely match the target set. As would be expected, larger population sizes can test more search arguments, which results in higher top fitness values.

Figure 6b shows the average number of generations required for convergence. Here, we see that the population size has a large impact on the number of generations required for convergence. Larger population sizes tend to converge with fewer generations, although this benefit can easily be compromised by the longer computational time required. While Figure 6b shows the average generations to convergence, Table I shows the actual percentage of tests which were able to find a solution with a fitness of 0.99 within 10,000 generations. We see again that larger population sizes have higher convergence rates over this generation limit.

B. Global Transformation

For the second experiment, each test set is given a random transformation of translation, scale, and rotation. We again evaluate the average performance of the HoF GA over 10 runs using the same database of buildings of 30 nearest neighbors with a mutation probability of 1 and population sizes of 50, 100, 250, 500 and 1000. A fitness value of 0.99 is required within 10,000 generations for convergence.

TABLE I
PERCENT OF TESTS WHICH CONVERGE WITHIN 10,000 GENERATIONS

Population Size	Resubstitution	Global Transformation
50	55.0%	34.4%
100	70.7%	43.2%
250	84.5%	52.7%
500	90.1%	58.1%
1000	94.8%	60.5%

Convergence is defined as having at least one chromosome in the population with a fitness value of 0.99 or greater. Tests using strict resubstitution have a higher convergence rate than those using global transformations. Larger population sizes are also shown to improve the convergence rate.

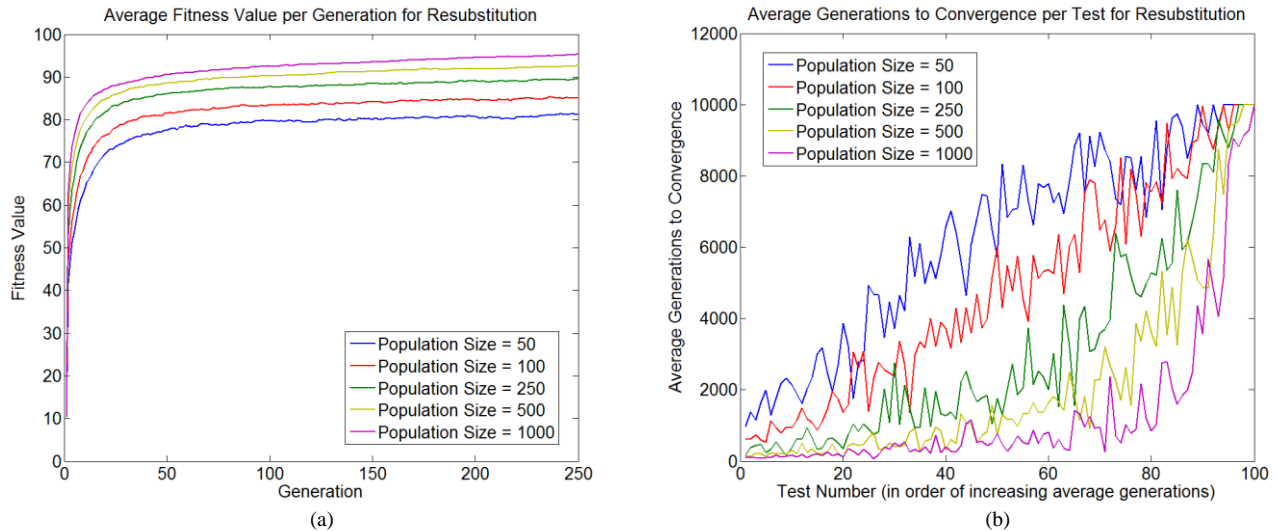


Fig. 6 (a) Average fitness value per generation for resubstitution. The average top fitness value within the population rapidly increases during the first 50 generations and then begins to level out. Larger population sizes result in higher top fitness values per generation. (b) Average generations to convergence per test for resubstitution. Generations are limited to 10,000 and convergence is defined by the population having at least one chromosome with a fitness value of 0.99 or greater. Larger population sizes result in fewer average generations required to reach convergence.

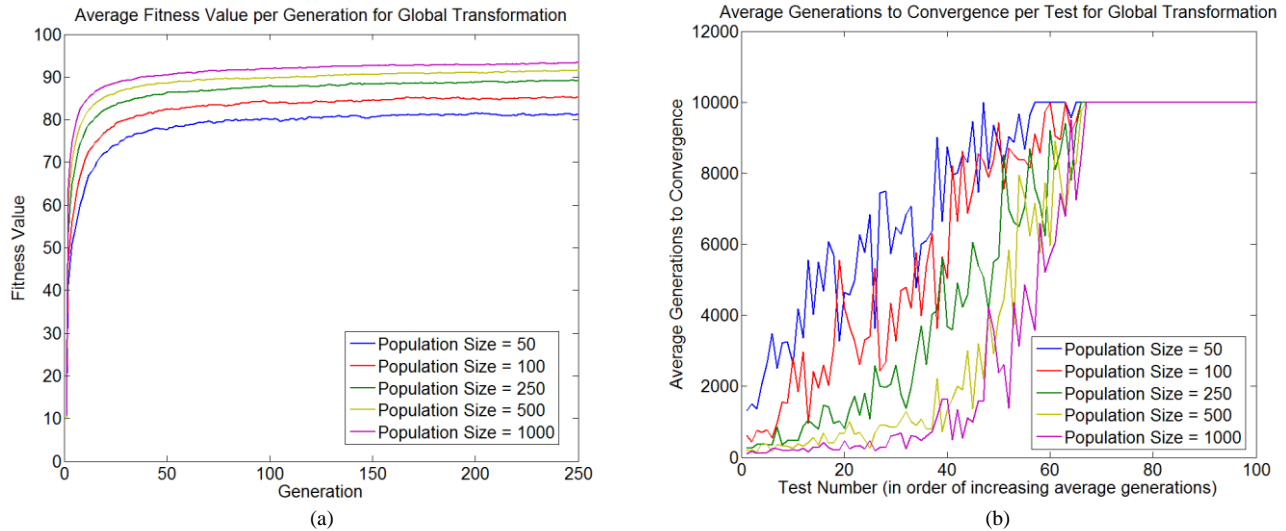


Fig. 7 (a) Average fitness value per generation for global transformation. The average top fitness values are only slightly lower than those of resubstitution. Again, larger population sizes result in higher top fitness values. (b) Average generations to convergence per test for global transformation. Generations are limited to 10,000 and convergence is defined by the population having at least one chromosome with a fitness value of 0.99 or greater. Many tests required more generations to converge than with resubstitution and a significant number of tests failed to converge at all. Larger population sizes again result in fewer average generations required to reach convergence.

Figure 7a shows the average top fitness values per generation, which looks very similar to the resubstitution results. Figure 7b, however, shows that many of the tests required more generations to converge, and a significant number of them failed to converge at all within 10,000 generations. Table I confirms that the percentage of tests which converge is much lower for the global transformation experiment.

Despite the slow convergence times for global transformations, the average top fitness values per generation are only slightly lower than those observed for resubstitution. This shows that although the perfect match is not found quickly, the top matches still reflect the spatial configuration shown in the input sketch. While the histograms of forces themselves should account for global changes in translation and scale, a global rotation results in a shifting of the histo-

grams. This is accounted for in the matching algorithm, but clearly rotation has an impact on the performance of the GA. The fitness value of 0.99 required for convergence may also be too strict due to rounding errors accumulated from the extra rotation required.

C. Practical Example

To test our algorithm with an output sketch from the T₂S system, we use the sketch from Figure 1, which describes the movement of a person moving north along a road in downtown Columbia, MO. The sketch was generated automatically from a lengthy text description [1] and consists of eight buildings with rectangular or L-shaped outlines. The sketch also contains information about parking lots and roads which our algorithm does not utilize at this time. Us-

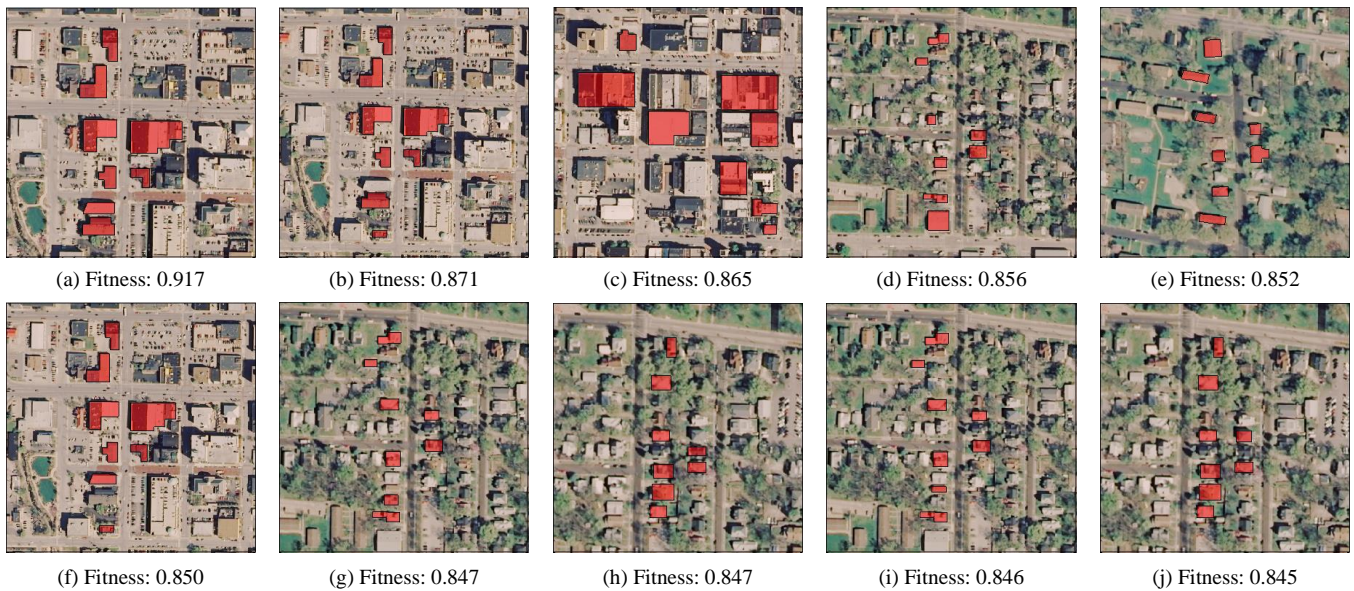


Fig. 8 Top ten results returned by the HoF GA given the input sketch in Figure 1. Results (a), (b), and (f) describe the same location described by the input sketch with result (a) corresponding to the actual buildings used. Note that the fitness value is not perfect due to the generalized building shapes in the input sketch. Results (d), (e), (g), (h), (i), and (j) all describe residential locations that could be eliminated with additional search criteria. Result (c) shows the rotation invariance of the algorithm with the buildings described all rotated roughly 45° . Again, additional search criteria such as road networks could eliminate this as a potential result.

ing this sketch required a database with connections to 100 nearest neighbors. Figure 8 shows the top results returned by the HoF GA using a population size of 50 chromosomes.

Three of the top ten results describe the same location described by the input sketch with the top result corresponding to the actual buildings used. The fitness of the top result is 0.917, which is not a perfect fitness of 1 due to the generalized building shapes used in the input sketch. Six of the top ten results described residential locations that could be eliminated from the results with some additional search criteria such as the original environment of the input sketch. The result shown in Figure 8c is of particular interest because it shows the buildings rotated roughly 45° from the input sketch. Since the original sketch clearly shows a grid-like road structure, any rotation that is not a multiple of 90° in an urban setting is unlikely to be correct. Because only the histograms between buildings were compared, the match has a fitness value of 0.865 despite its unlikely arrangement.

These results show the variety of configurations that can match a sketch. They all have buildings placed in the proper relative locations, but their shapes and sizes vary considerably. This shows that the placement of objects in a sketch is typically a greater factor in finding a good match than exact building shapes or sizes. Additional search criteria, such as the parking lots or road infrastructure shown in the sketch, would likely improve the search results.

V. CONCLUSION

In this paper, we developed a modification to the traditional GA for the purpose of building matching. This algorithm uses a special mutation operator which applies the histograms of forces to select an ideal set of buildings. We

tested our algorithm on several sets of buildings using strict substitution and also with a global transformation. We found that even though a global rotation of the input sketch can take longer to converge to a single solution, the results obtained in both cases reflect the spatial locations of the original buildings more than their specific shapes. This demonstrates the usefulness of using HoF relations when comparing spatial relationships. The algorithm's invariance to global rotation, however, can lead to matches that defy common sense, such as rotations of 45° in a grid-like urban setting. Future work in this area will explore the effect of imposing additional search criteria such as road networks and specific object types (parking lots, restaurants, etc.).

VI. REFERENCES

- [1] I. Sledge and J. Keller, "Mapping Natural Language to Imagery: Placing Objects Intelligently," *IEEE Proceedings, International Conference on Fuzzy Systems (FUZZ-IEEE)*, Jeju Island, Korea, August, 2009, pp. 518-524.
- [2] P. Matsakis, J. M. Keller, O. Sjahputera, and J. Marjamaa, "The Use of Force Histograms for Affine-Invariant Relative Position Description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1-18, Jan. 2004.
- [3] A. P. Engelbrecht, *Computational Intelligence: An introduction*, 2nd ed. Chichester, West Sussex: John Wiley, 2007, pp. 125-176.
- [4] J. H. Holland, "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor, 1975.
- [5] J. R. Beveridge, "Optimal 2D Model Matching Using a Messy Genetic Algorithm," *Proceedings, American Association for Artificial Intelligence*, pp. 677-683, 1998.
- [6] M. A. Rodríguez and M. C. Jarur, "A Genetic Algorithm for Searching Spatial Configurations," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, June 2005.
- [7] P. Matsakis and L. Wendling, "A new way to represent the relative position between areal objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 634-643, 1999.