An Evolutionary Framework for Matching Geospatial Object Configurations

Presented by Andrew Buck

Committee Members

Dr. James Keller
Dr. Marjorie Skubic
Dr. Mihail Popescu

April 27, 2012

University of Missouri
Department of Electrical and Computer Engineering

Outline

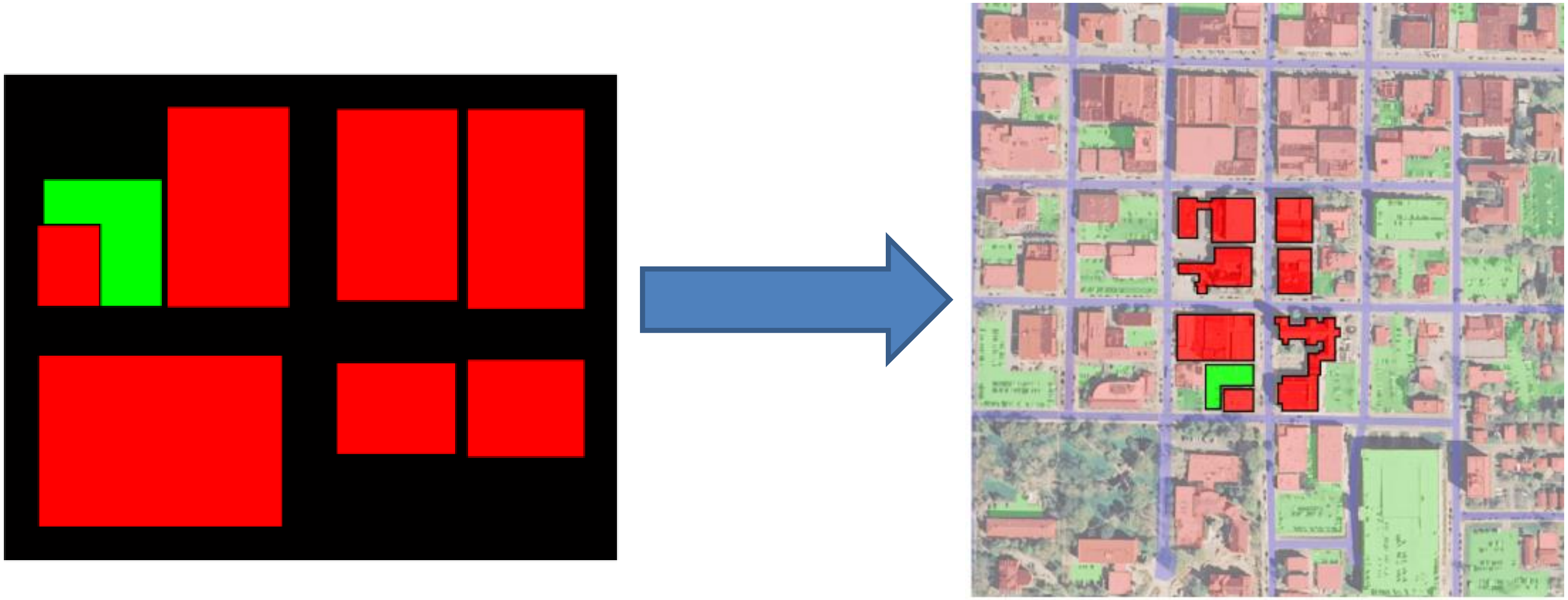
- Problem Overview
- Spatial Reasoning
  - – Histograms of Forces
  - – Representing Object Sets
- Design of the Matching Algorithm
  - – Comparing Object Sets
  - – Local Search Operators
- Experiments and Results

Problem Overview

Where am I?

> “I see a large building to my right and a small building on my left surrounded by a parking lot.”

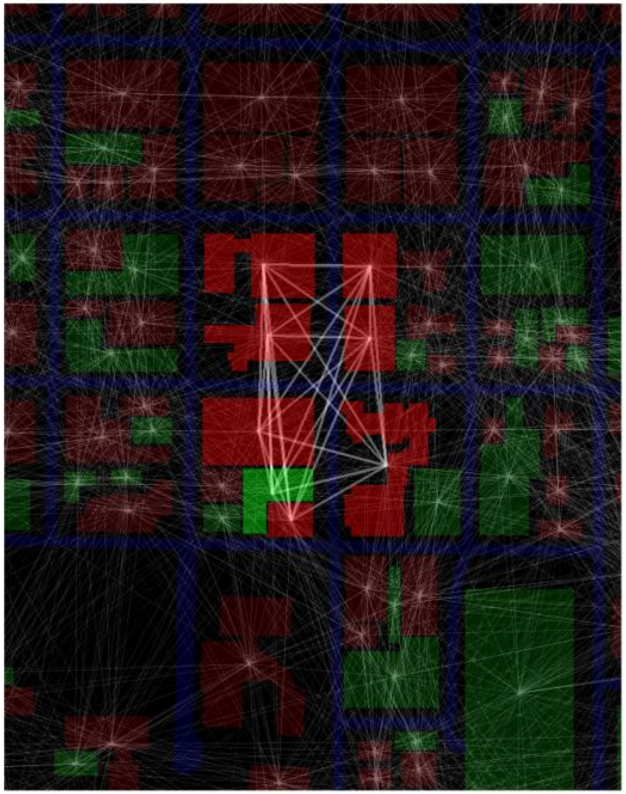
Can I draw a map?

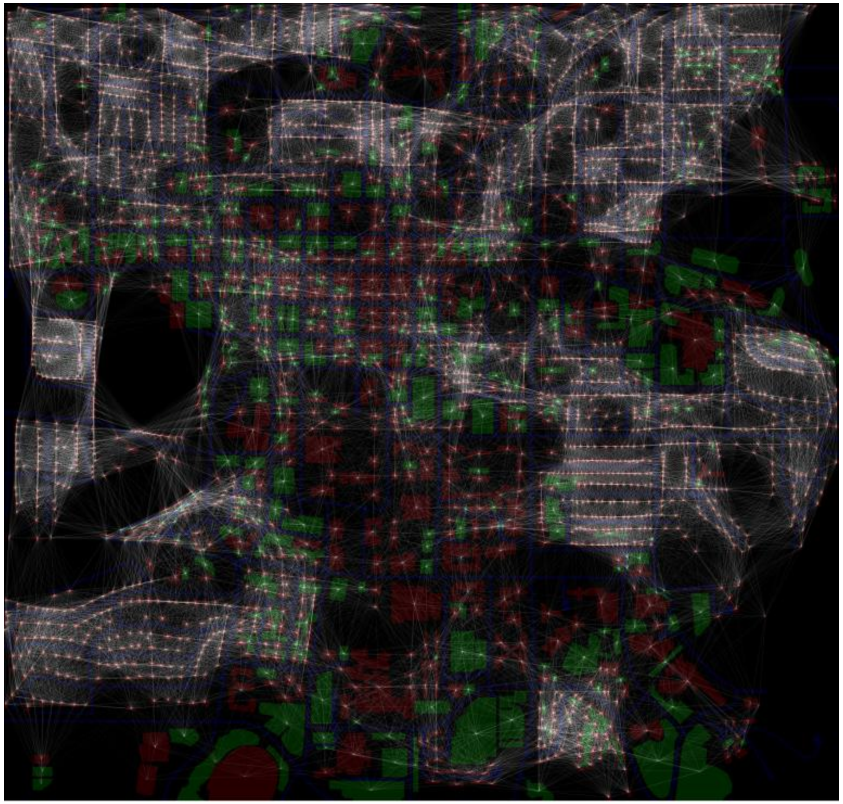


Problem Overview

How can I represent the sketch?

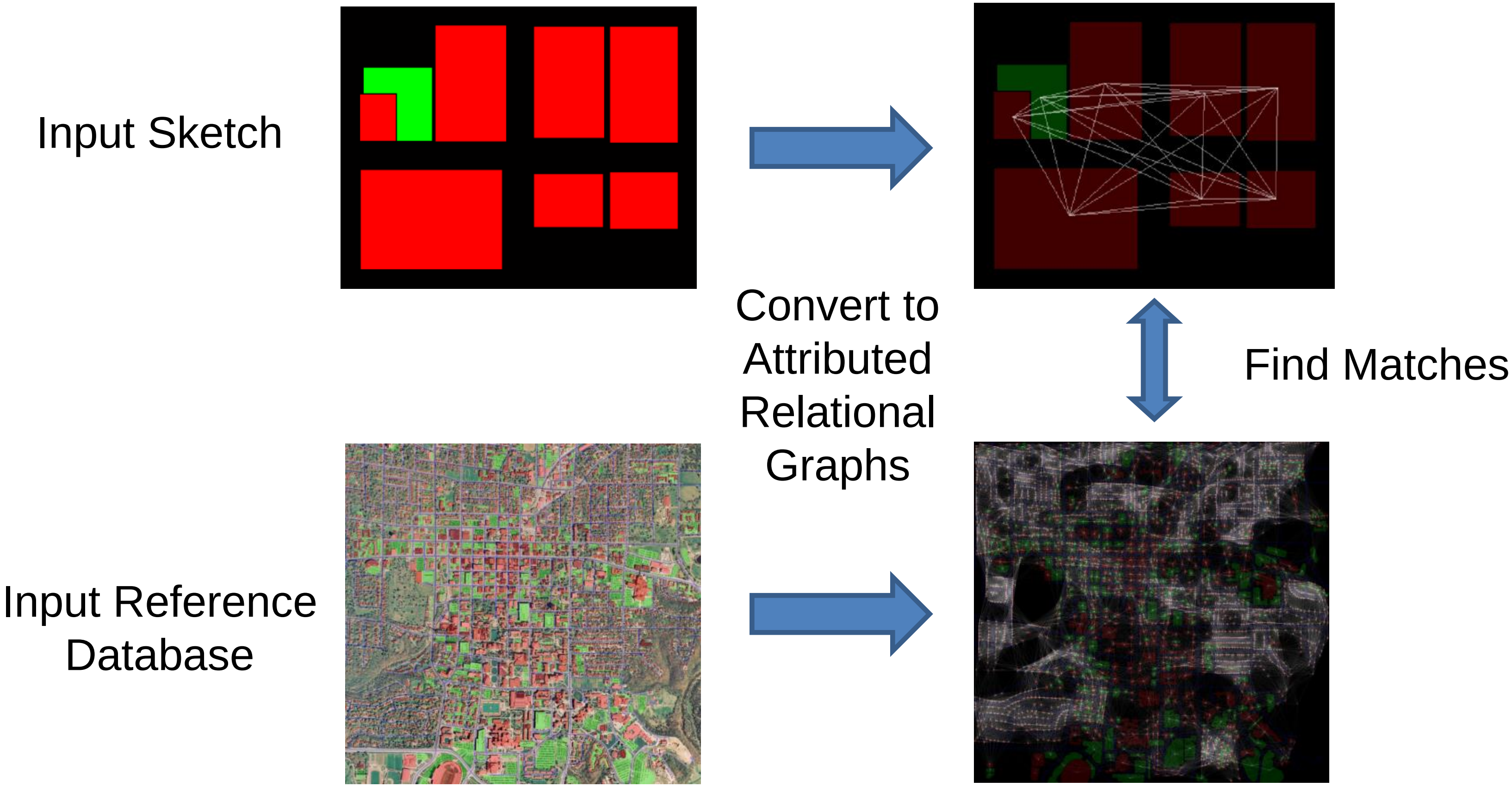
How can I search the database?





Problem Overview

Goal: Given an input sketch and a reference database, find the most likely location from which the sketch originated.



Input Sketch

Convert to Attributed Relational Graphs

Find Matches

Input Reference Database

Spatial Relationships

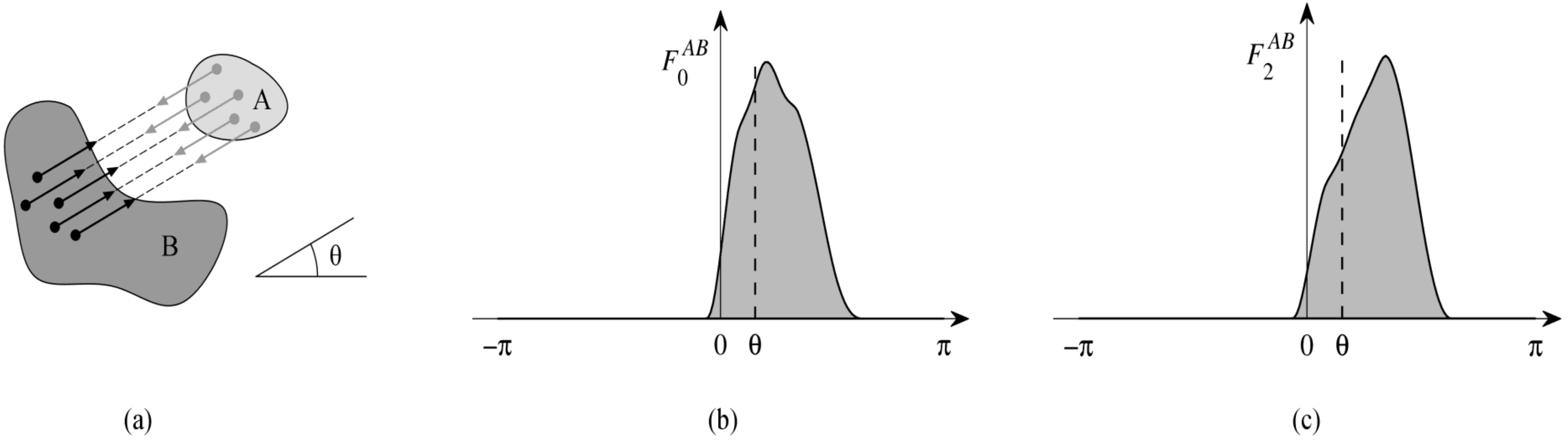
Given a set of objects, how can we describe their spatial organization?

- Direction
- Distance
- Size
- Shape
- Topology

Should these terms be crisp or fuzzy?

Histograms of Forces

A force histogram $F_r^{AB}(\theta)$ is a way of representing the degree of truth of the statement, “A is in direction θ from B.”



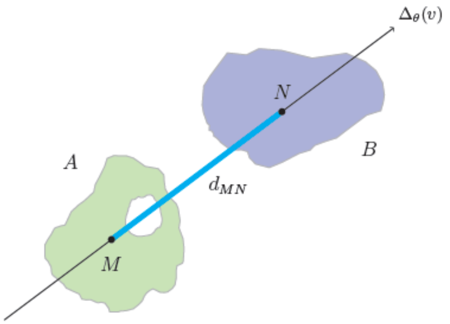
(a)

(b)

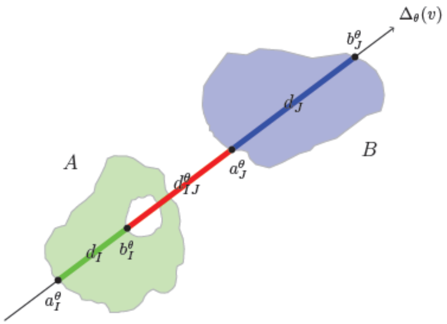
(c)

(a) A force histogram is the scalar resultant of elementary forces exerted by the points of A on those of B. Each one pulls B in direction θ. (b) The histogram of constant forces $(r = 0)$ is one representation that provides a global perspective. (c) The histogram of gravitational forces $(r = 2)$ is another possible representation, which is more sensitive to nearby points.

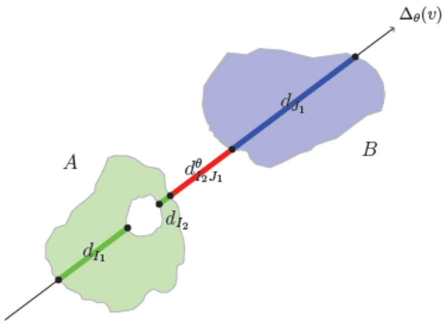
Histograms of Forces



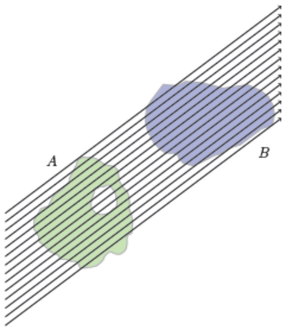
Points



Line Segments



Longitudinal Sections



Directions

Handling Points

$$\phi_r(M - N) = \frac{1}{d^r_{MN}}$$

Handling Line Segments

$$f_r\left(d_I, d^\theta_{IJ}, d_J\right) = \int_{a^\theta_I}^{b^\theta_I} \int_{a^\theta_J}^{b^\theta_J} \phi_r(u - v)\,dv\,du$$

Handling Longitudinal Sections

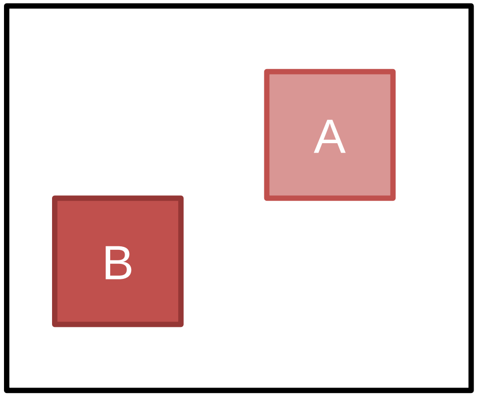
$$\mathcal{F}_r\left(\theta, A_\theta(v), B_\theta(v)\right) = \sum_{i,j} f_r\left(d_{I_i}, d^\theta_{I_iJ_j}, d_{J_j}\right)$$

Handling Directions

$$F^{AB}_r(\theta) = \int_{-\infty}^{\infty} \mathcal{F}_r\left(\theta, A_\theta(v), B_\theta(v)\right)dv$$

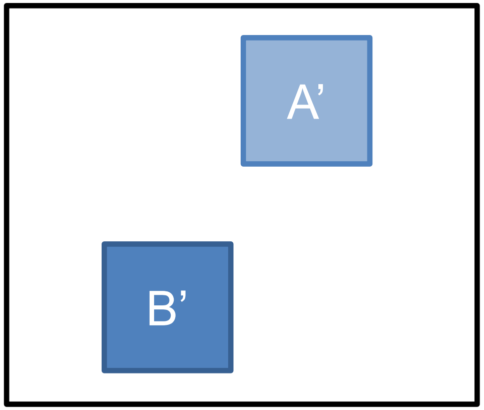
Comparing Force Histograms

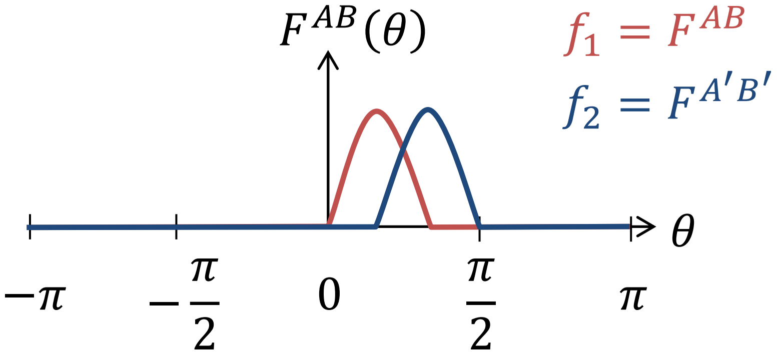
How do we compare force histograms?



Normalized Cross-Correlation:

$$\mu_C(f_1, f_2) = \frac{\sum_\theta f_1(\theta) f_2(\theta)}{\sqrt{\sum_\theta f_1^2(\theta)}\,\sqrt{\sum_\theta f_2^2(\theta)}}$$





Normalized Cross-Correlation

$$\mu_C(f_1, f_2) = \frac{\sum_\theta f_1(\theta) f_2(\theta)}{\sqrt{\sum_\theta f_1^2(\theta)}\sqrt{\sum_\theta f_2^2(\theta)}}$$

The normalized cross-correlation satisfies the following properties:

$$0 \leq \mu_C(f_1, f_2) \leq 1$$

$$f_1 = f_2 \Rightarrow \mu_C(f_1, f_2) = 1$$

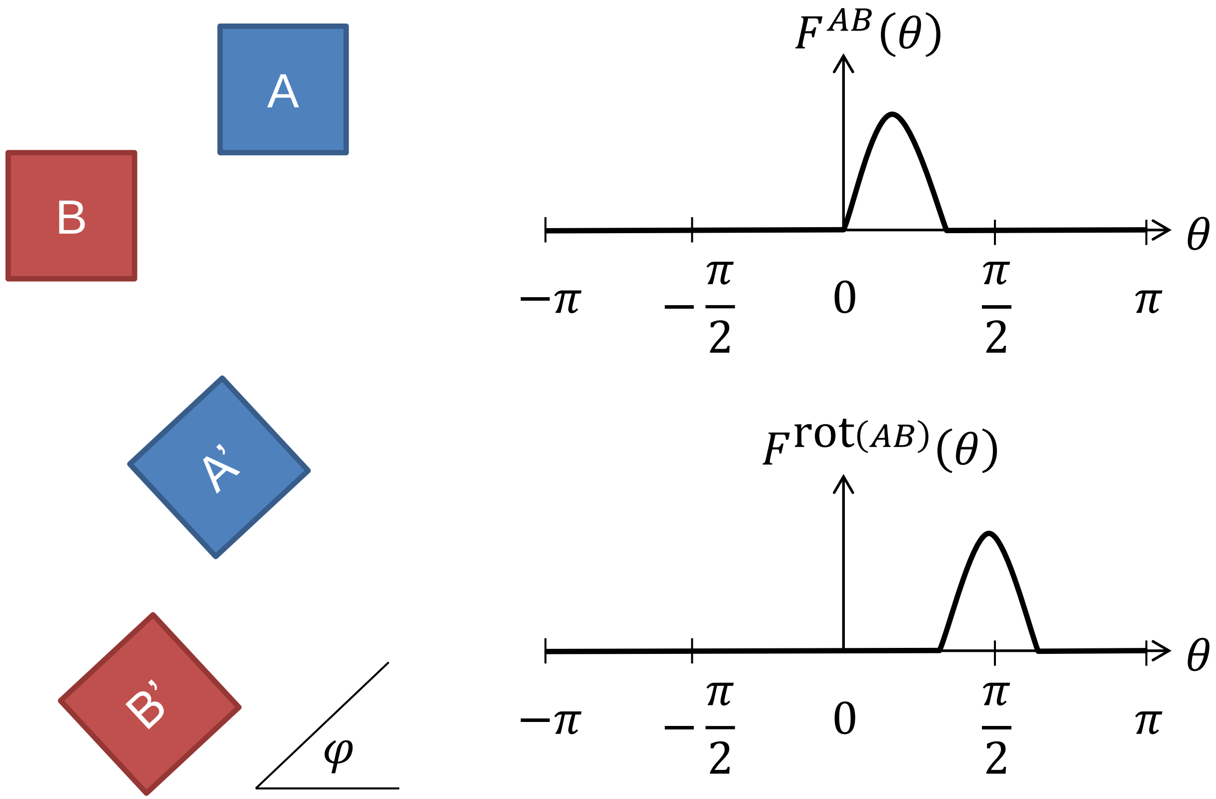
$$\mu_C(f_1, f_2) = \mu_C(f_2, f_1)$$

$$\forall \lambda_1 \in \mathbb{R}_+^*, \forall \lambda_2 \in \mathbb{R}_+^*, \qquad \mu_C(\lambda_1 f_1, \lambda_2 f_2) = \mu_C(f_1, f_2)$$

Effect of Rotation

Rotating a pair of objects is equivalent to shifting the force histogram.

$$F_r^{\mathrm{rot}(AB)}(\theta) = F_r^{AB}(\theta - \varphi)$$

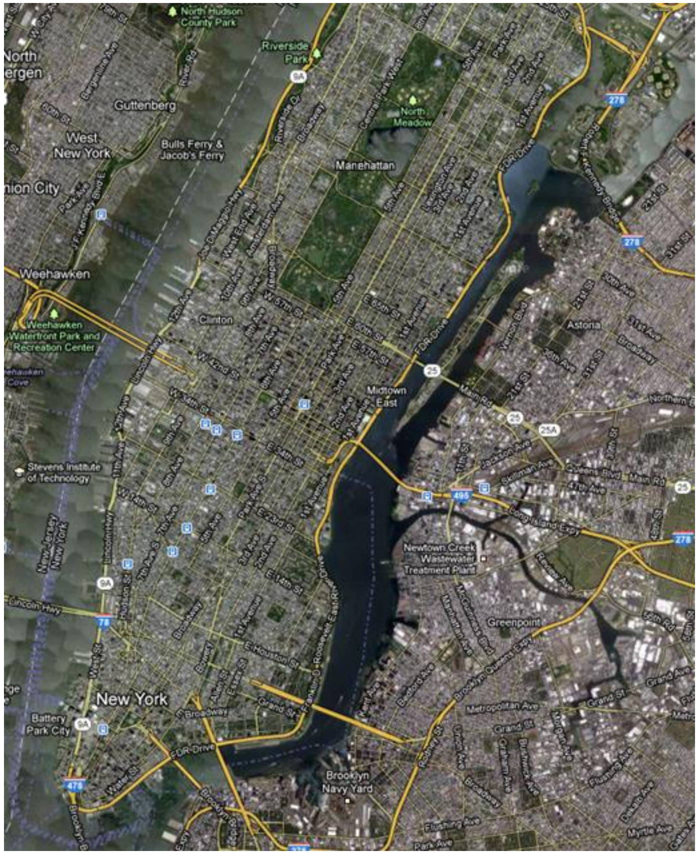


Towards Rotation Invariance

Ideally, we would like for the prior two sets to be considered equivalent.

This allows sketches to be drawn with any orientation.

This is particularly important for when hand-drawn maps are not aligned with the ground truth street grid, such as in Manhattan.



Comparing Pairs of Objects

Given two pairs of objects, (A, B) and (A', B'), defined with reference angles ϕ and ϕ' respectively, we can compare their relative spatial relationship with the general equation

$$\mu_{Pair}(A, B, \phi, A', B', \phi') = \beta\mu_{C0} + (1-\beta)\mu_{C2}, \qquad \text{where}$$

$$\mu_{C0} = \mu_C\left(F_0^{AB}(\theta-\phi), F_0^{A'B'}(\theta-\phi')\right),$$

$$\mu_{C2} = \mu_C\left(F_2^{AB}(\theta-\phi), F_2^{A'B'}(\theta-\phi')\right).$$

β is a weighting factor, typically set to 0.5 to give equal weight to both histograms.

Main Direction

To achieve rotational invariance, we need to shift each histogram to a common value.

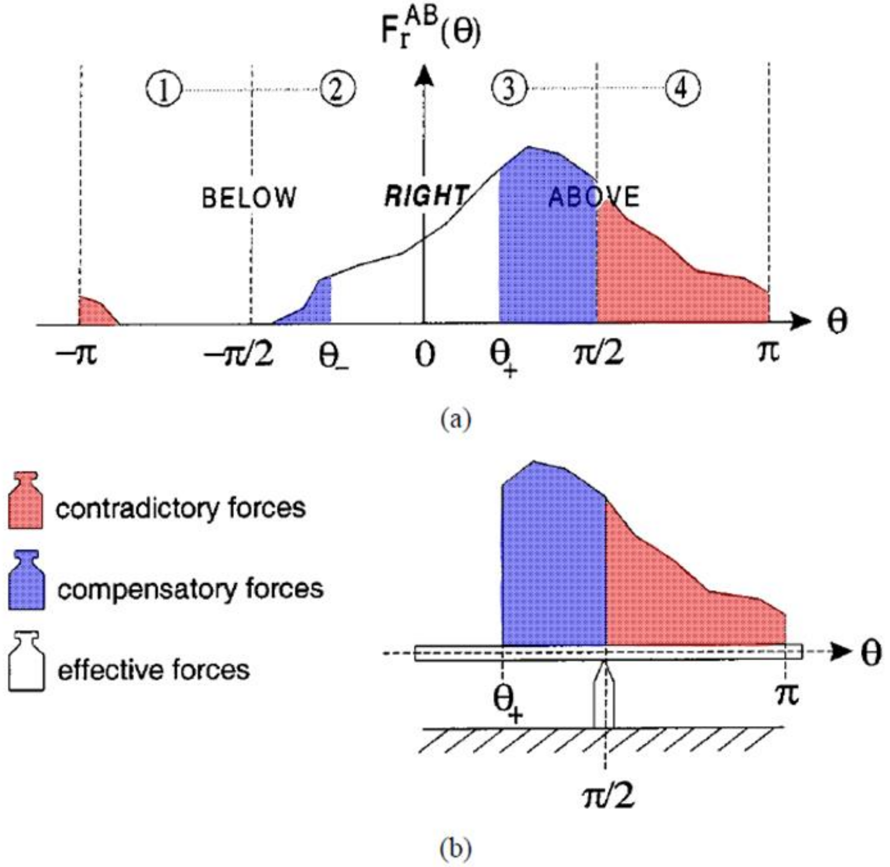
How far to shift?

For a force histogram $F^{AB}(\theta)$ we define a single *main direction* φ^{AB} which best captures the directional relationship between A and B.

How to pick φ^{AB}?

Main Direction

For a given force histogram F_r^{AB} and angle θ, divide the forces into effective, contradictory, and compensatory forces.



(a)

(b)

Define $b_r^{AB}(\theta)$ as the percentage of forces which are effective (maximum degree of truth).

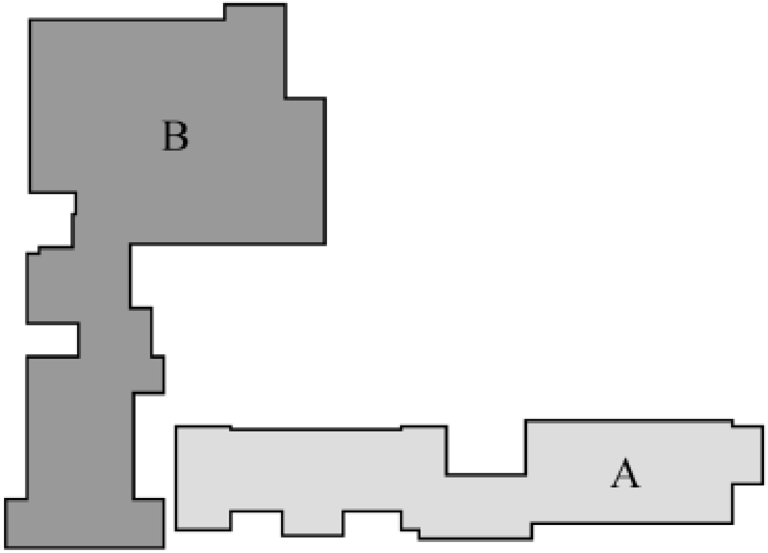
Define $a_r^{AB}(\theta)$ as $b_r^{AB}(\theta)$ times the degree to which the center of mass of the effective forces aligns with $\theta = 0$ (effective degree of truth).

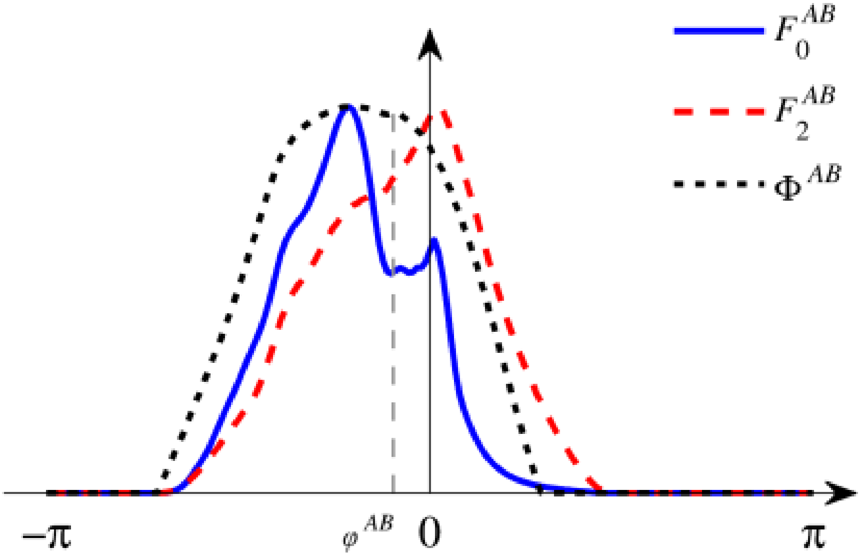
[Matsakis, et al., 2001]

Main Direction

We should balance the contributions of the constant and gravitational forces. Define the main direction histogram as

$$\Phi^{AB}(\theta) = \max\{a_0^{AB}(\theta), \min\{a_2^{AB}(\theta), b_0^{AB}(\theta)\}\}$$

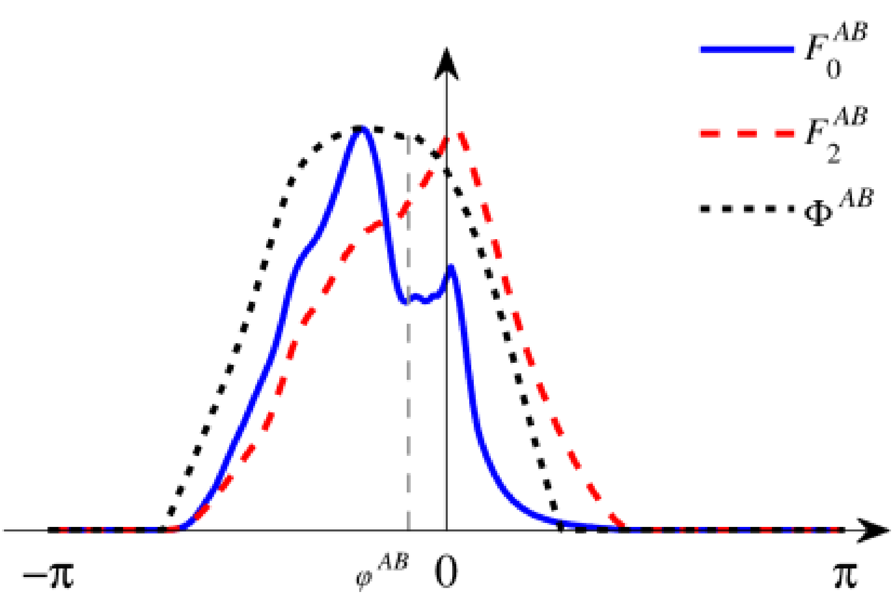


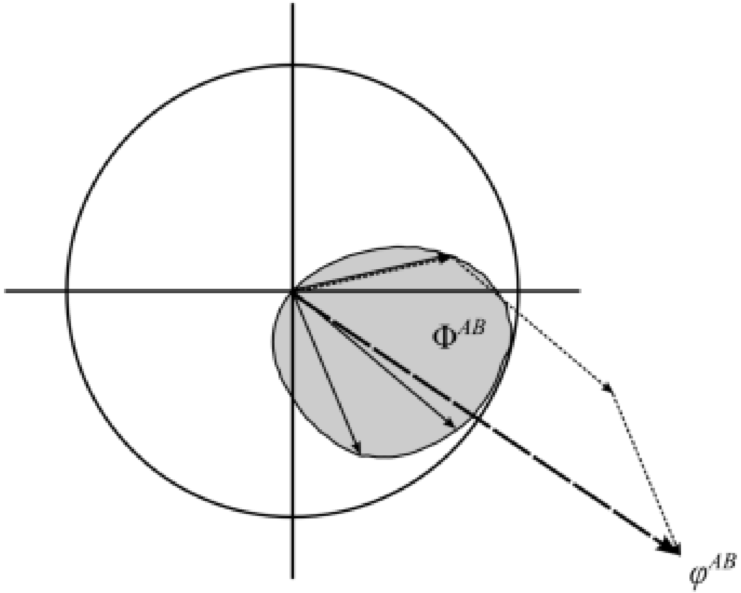


Main Direction

Compute the main direction φ^{AB} as the centroid of $\Phi^{AB}(\theta)$. This requires polar vector summation.

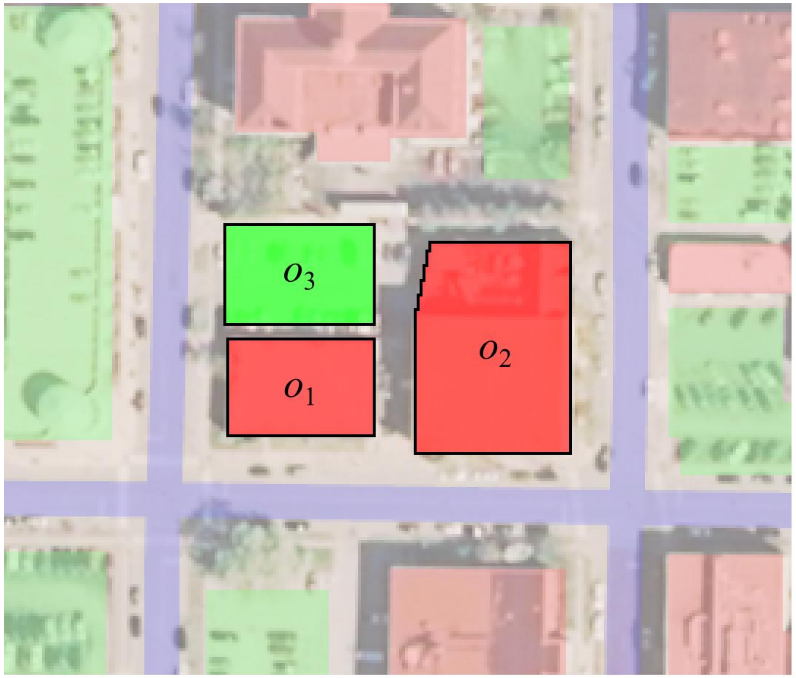
$$\varphi^{AB} = \operatorname{atan2}\left(\sum_{\theta\in[0,2\pi]} \sin\left(\Phi^{AB}(\theta)\right), \sum_{\theta\in[0,2\pi]} \cos\left(\Phi^{AB}(\theta)\right)\right)$$

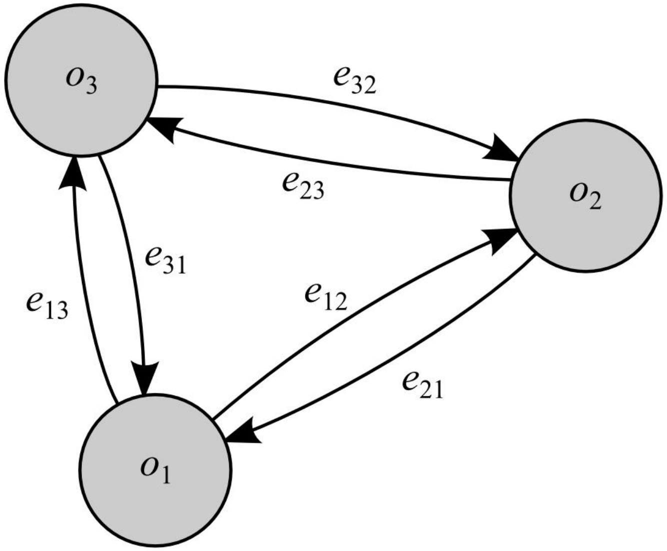




Representing Object Sets

Given a scene consisting of a set of 2D objects, $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$, each object becomes a node in an attributed relational graph (ARG) and the relationships between objects become edges.





$\mathcal{O} = \{o_1, o_2, o_3\}$ $E_{\mathcal{O}} = \{e_{12}, e_{13}, e_{21}, e_{23}, e_{31}, e_{32}\}$, where $e_{ij} = (o_i, o_j) \in \mathcal{O} \times \mathcal{O}$

Representing Object Sets

Assign attributes to the nodes and edges. Node attributes are object types and edge attributes are sets of force histograms.

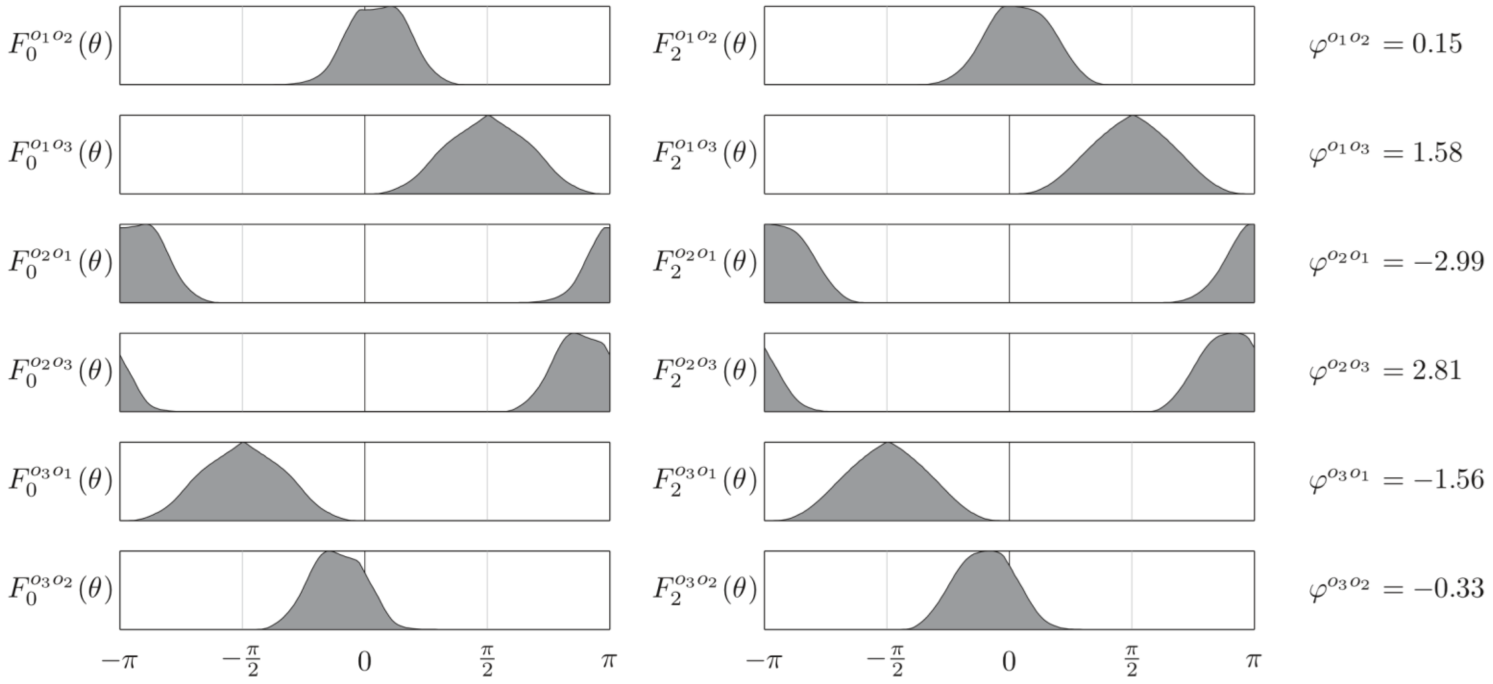
$\mathcal{L} = \{\text{"building"}, \text{"parking lot"}\}$

$L_{\mathcal{O}}(o_1) = L_{\mathcal{O}}(o_2) = \text{"building"}$

$L_{\mathcal{O}}(o_3) = \text{"parking lot"}$

$H_{\mathcal{O}} = \{h_{12}, h_{13}, h_{21}, h_{23}, h_{31}, h_{32}\}$,
where $h_{ij} = \left(F_0^{o_i o_j}, F_2^{o_i o_j}, \varphi^{o_i o_j}\right)$

Full ARG: $G_{\mathcal{O}} = (\mathcal{O}, E_{\mathcal{O}}, L_{\mathcal{O}}, H_{\mathcal{O}})$



Graph Size

A completely defined ARG for a set of n objects will have n vertices and $n \times (n-1)$ edges. For large graphs, such as the reference database, only the closest objects have edge relationships.

The size of the ARG can cut in half by only considering edges for which $i < j$. This is possible due to the semantic inverse property of the histograms of forces.

$$F_r^{BA}(\theta) = F_r^{AB}(\theta + \pi)$$

Problem Definition

Given:

Reference Set: $\mathcal{R} = \{x_1, x_2, \dots, x_m\}$

Sketch: $S = (o_1, o_2, \dots, o_n)$, $\quad$ where $m \gg n$

Goal:

Find a mapping function $\Gamma: S \rightarrow \mathcal{R}$ represented as the candidate set $\Gamma = \left(x_{(1)}, x_{(2)}, \dots, x_{(n)}\right)$, $\; x_{(i)} \in \mathcal{R}$ such that $\Gamma(o_i) = x_{(i)}$.

The mapping function should preserve the object labels and spatial organization of the sketch.

Comparing Object Sets (No Rotation)

If both S and Γ are defined with the same orientation, their similarity can be defined as

$$\Psi_1(S,\Gamma) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mu_{Pair}\left(o_i, o_j, 0, x_{(i)}, x_{(j)}, 0\right).$$

Both reference angles are defined as 0, resulting in no shifting.

Complexity is $O(n^2\omega)$ where n is the number of objects in each set and ω is the number of force histogram bins.

Optimal Rotation Angle

When S and Γ have different orientations, we rotate the sketch by some optimal angle, similar to orienting a map.

The list of angular differences represents the total mismatch between the orientations of S and Γ.

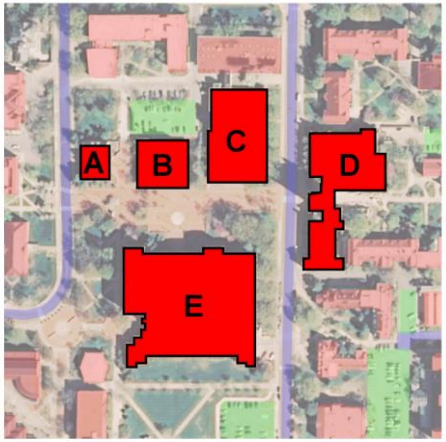
$$D = \{d_{11}, d_{12}, \ldots, d_{ij}, \ldots, d_{(n-1)n}\}, \qquad d_{ij} = \varphi^{o_i o_j} - \varphi^{x_{(i)} x_{(j)}}$$

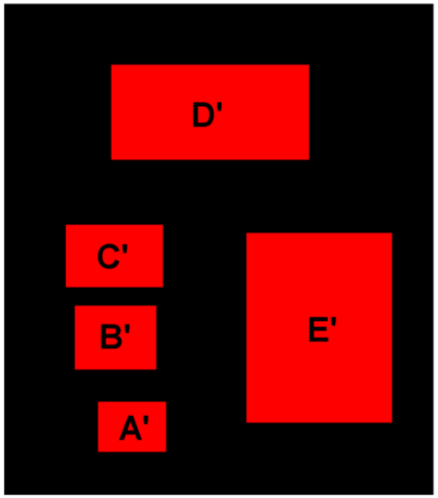
Pick the optimal rotation angle $\varphi^\star$ as the median angle of D.

$$\varphi^\star$$

$$= \arg\min_{d_{uv} \in D} [q(d_{uv})], \qquad q(d_{uv}) = \pi - \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left| \pi - \left| d_{ij} - d_{uv} \right| \right|$$

Optimal Rotation Angle





Object Pair		Main Direction (Ground Truth)	Main Direction (Sketch)	Angular Difference	Angular Distance
u	v	φ^{uv}	$\varphi^{u'v'}$	d_{uv}	$q(d_{uv})$
A	B	3.12	4.87	4.53	-25.39
A	C	3.30	4.81	4.78	-27.32
A	D	3.04	4.49	4.83	-27.49
A	E	2.12	3.58	4.83	-27.49
B	C	3.42	4.72	4.98	-26.73
B	D	3.02	4.36	4.94	-27.03
B	E	1.74	3.18	4.84	-27.48
C	D	2.77	4.24	4.81	-27.46
C	E	1.32	2.84	4.76	-27.21
D	E	0.62	1.99	4.91	-27.21

$\min(q) = -27.49$

$\varphi^{\star} = 4.83 \approx -83°$

Comparing Object Sets (With Rotation)

The similarity of S and Γ can now be defined as

$$\Psi_2(S,\Gamma) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mu_{Pair}\left(o_i, o_j, \varphi^\star, x_{(i)}, x_{(j)}, 0\right).$$

The sketch is rotated by the optimal rotation angle and the reference set remains unchanged.

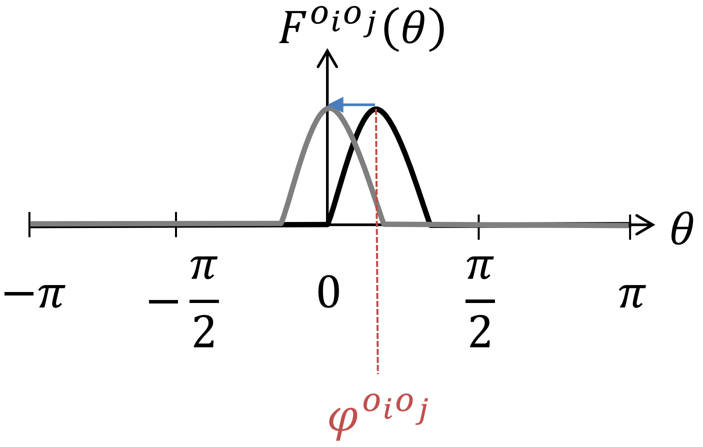
Elastic Angles

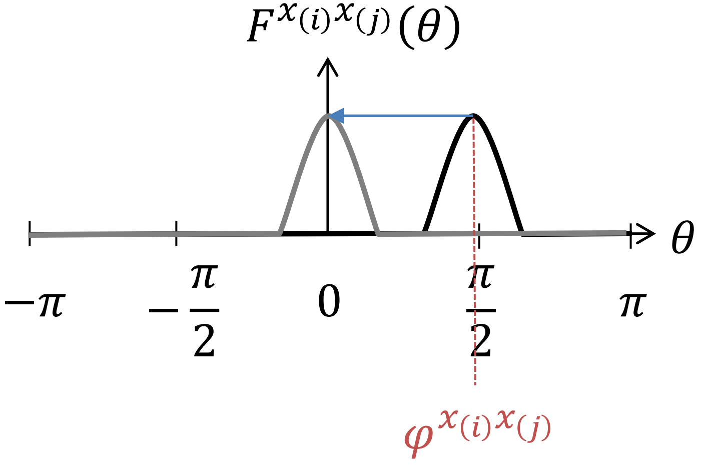
Rather than rotate the entire sketch by the same angle, each force histogram can be shifted individually.

This allows for small imperfections between the two object sets.

Each pair of force histograms is aligned by their main direction and multiplied by a weighting factor, based on how well they correspond to the previously calculated optimal rotation angle.

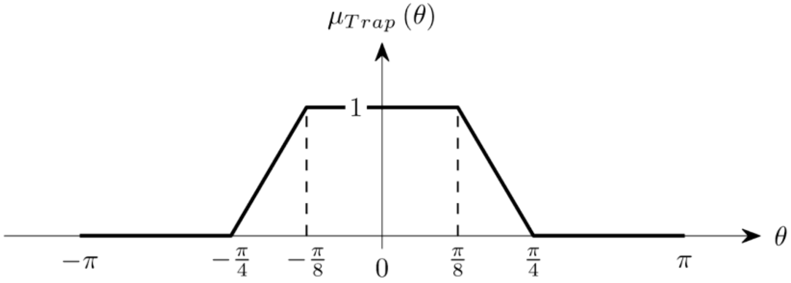
Elastic Angles





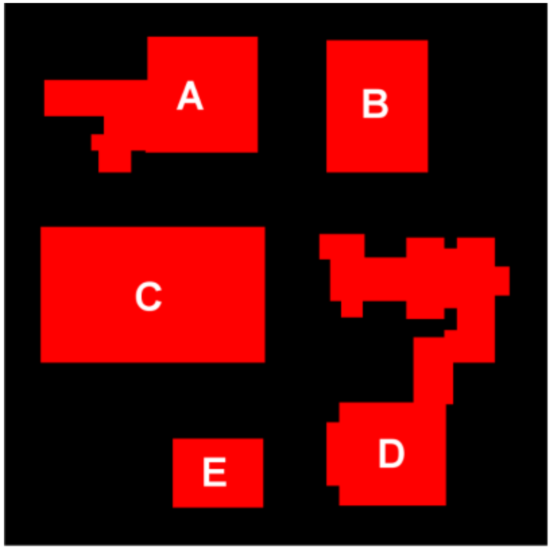
$$d_{ij} = \varphi^{o_i o_j} - \varphi^{x_{(i)} x_{(j)}}$$

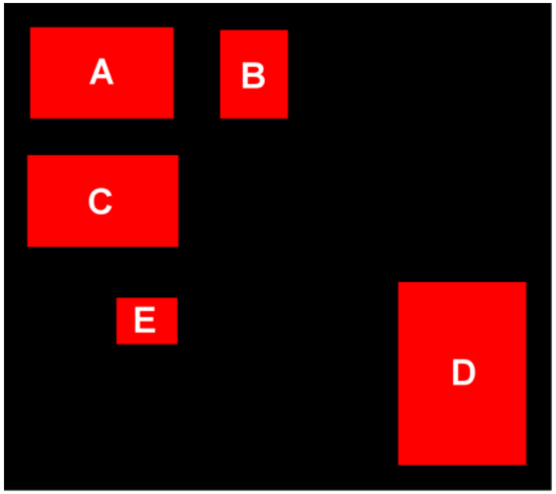
$\varphi^\star$ calculated as before

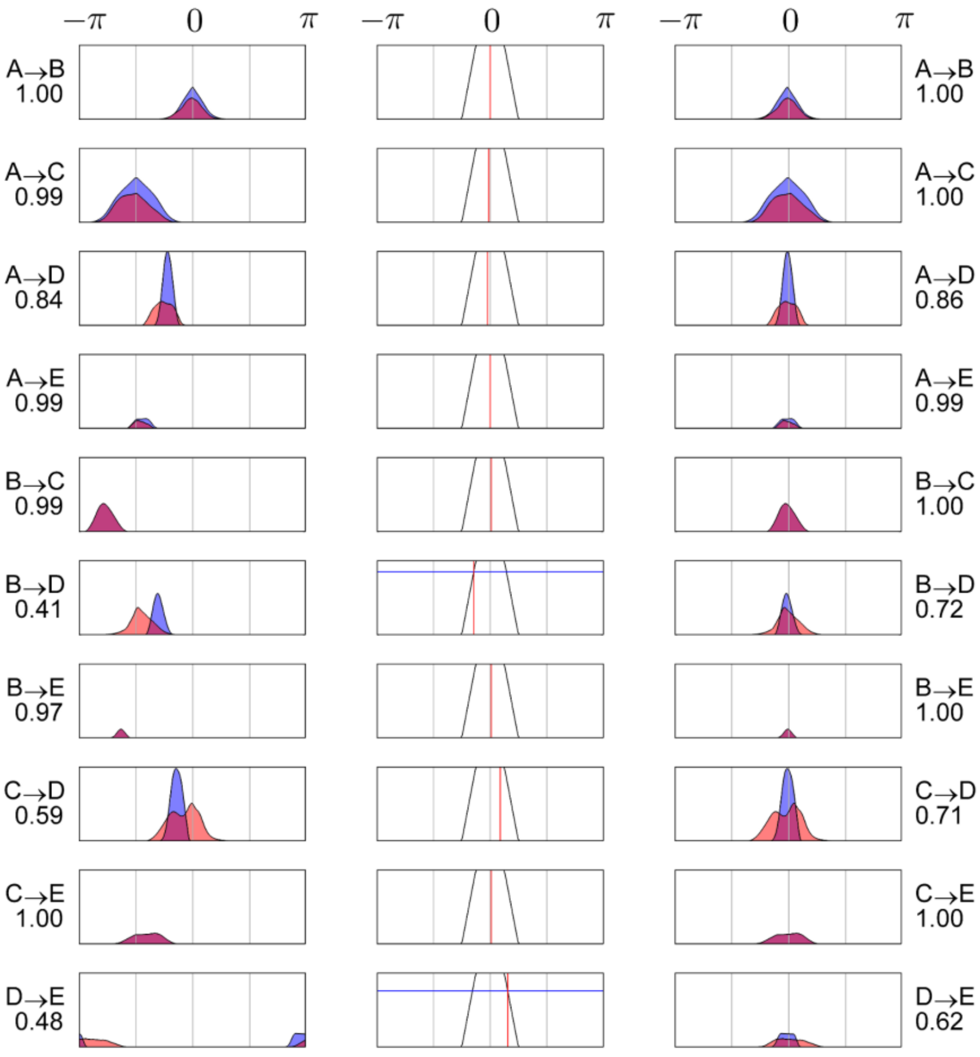


$$\Psi_3(S, \Gamma) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mu_{Trap}\left(\varphi^\star - d_{ij}\right) \times \mu_{Pair}\left(o_i, o_j, \varphi^{o_i o_j}, x_{(i)}, x_{(j)}, \varphi^{x_{(i)} x_{(j)}}\right)$$

Elastic Angles





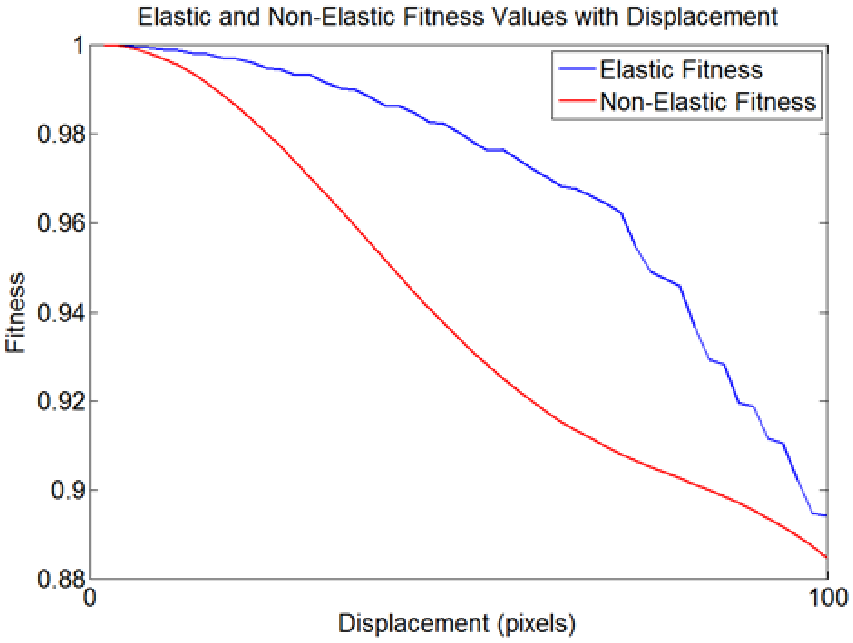


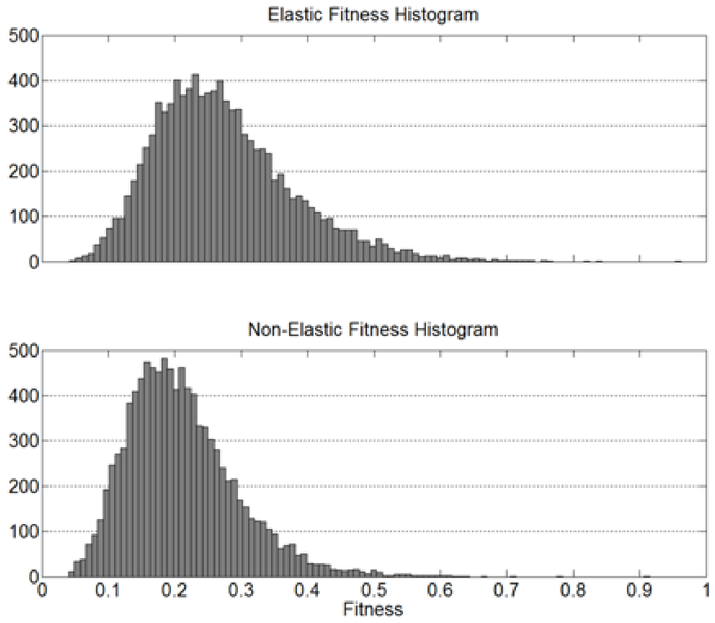
Non-Elastic Similarity: 0.83 $\mu_{Trap}(\theta)$ Elastic Similarity: 0.89

Elastic Angles

Elastic angles typically increases individual fitness values.

It does this for all sets of objects, raising the overall average fitness values.



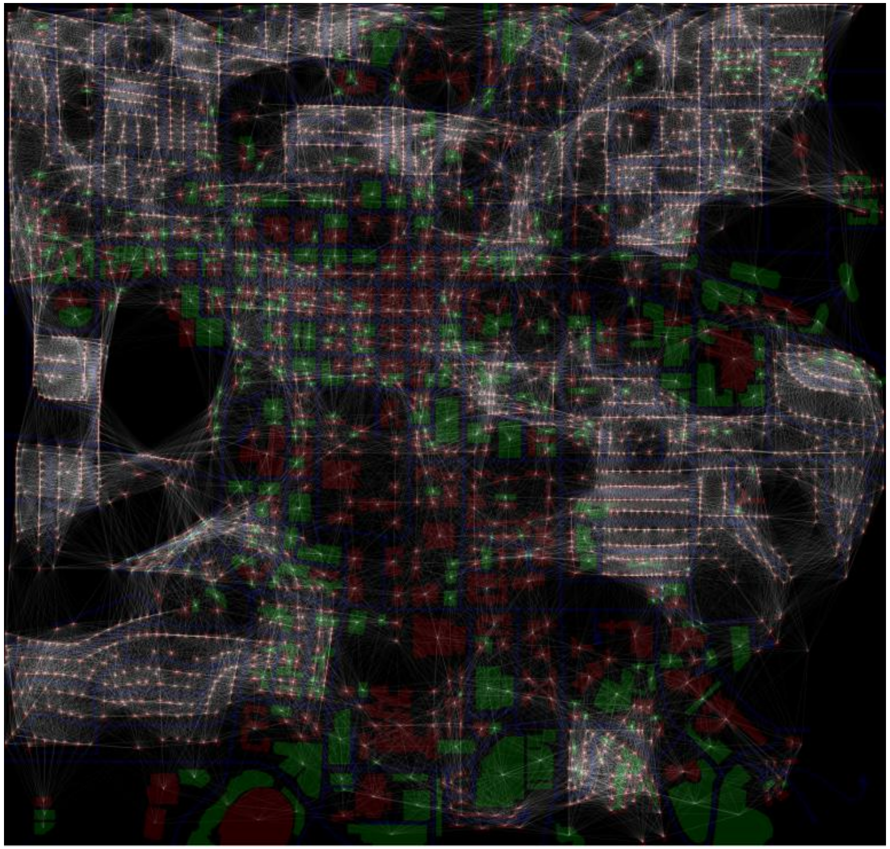


Reference Set Used for Testing

The reference set used in our experiments consists of 2814 buildings and parking lots from downtown Columbia, MO and the University of Missouri campus.

The graph contains edge relationships for a maximum of 50 nearest neighbors for each object.

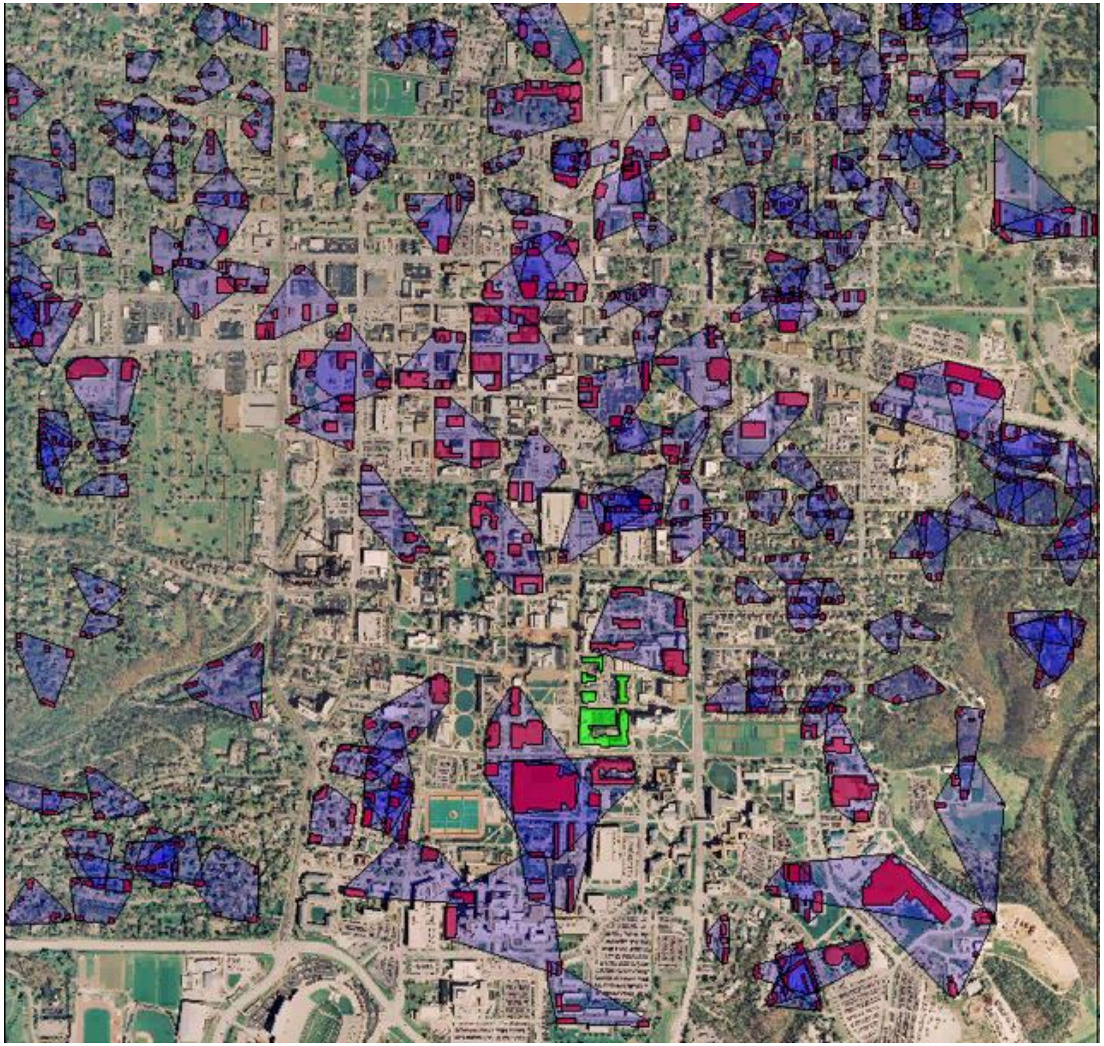
Objects which are too far apart or have too many other objects in between them are pruned from the graph.



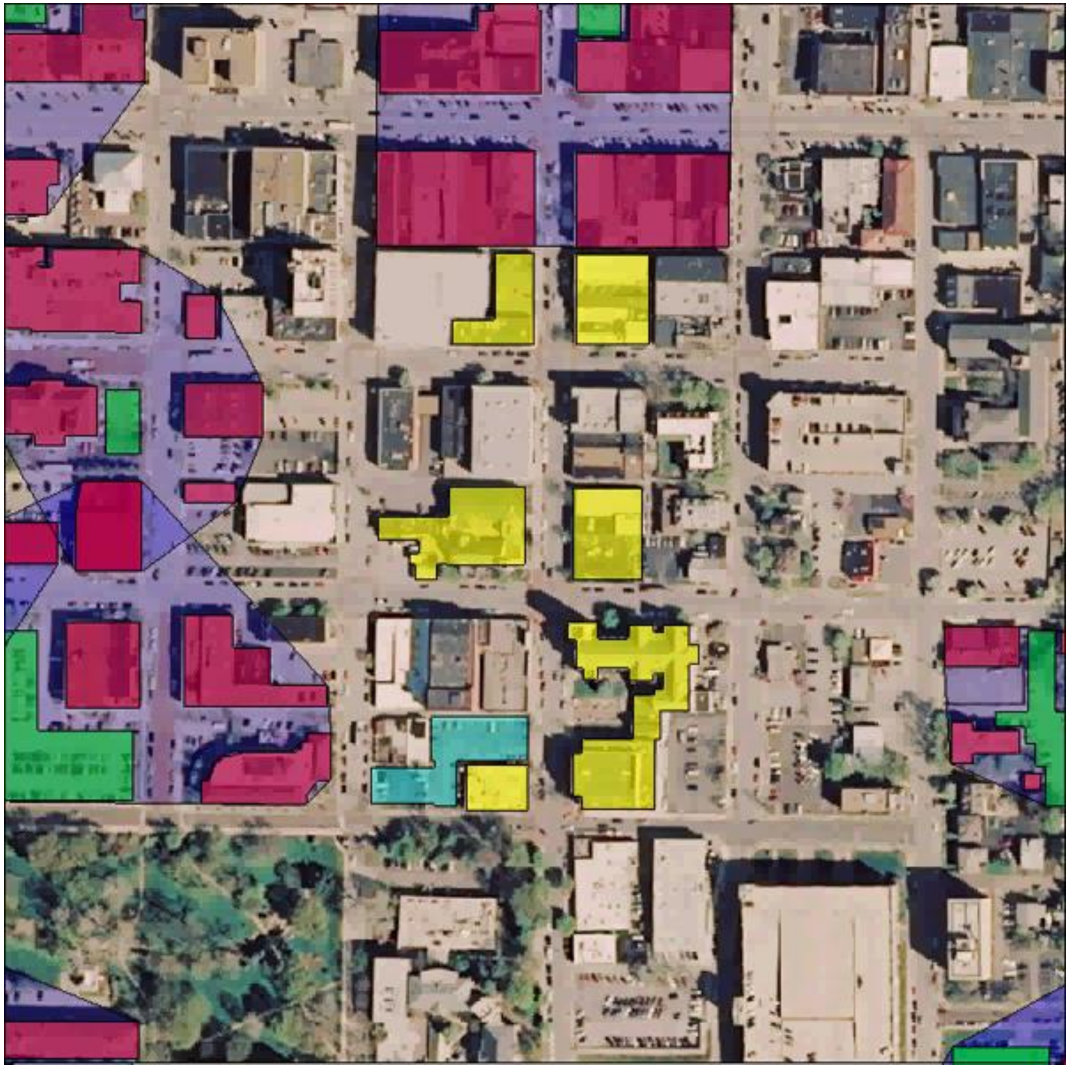
Outline of the Search Algorithm

1. Set $t = 0$ and create an initial population of random individuals: $P^{(0)} = \left(\Gamma_1, \Gamma_2, \ldots, \Gamma_\eta\right)$
2. For each individual in the population, $\Gamma_P \in P^{(t)}$
    a. Generate a set of possible children through mutation: $\mathcal{C} = \text{mutate}(\Gamma_P)$
    b. Select the most fit child $\Gamma_C = \underset{\Gamma \in \mathcal{C}}{\arg\max}\, \psi(\Gamma)$
    c. If the child is more fit than the parent, replace Γ_P with Γ_C
3. Every few generations, replace the lowest scoring individuals in $P^{(t)}$ with new random individuals
4. If stopping criteria is not met, increment t and go to 2
5. Return top scoring individual in $P^{(t)}$

Overview of the Search Algorithm



Close-Up of the Final Convergence



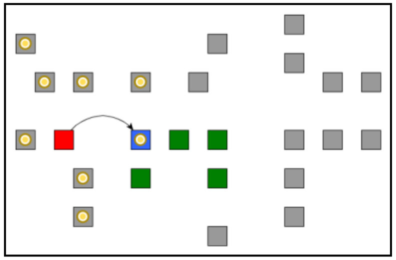
Mutation Operators

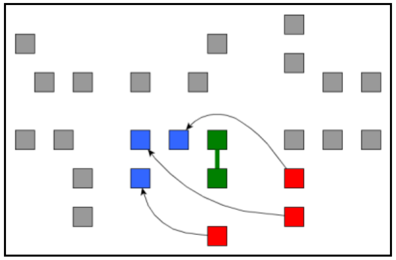
Mutation operators work as local search methods, seeking to improve the quality of each solution.

The local search is applied repeatedly to each individual in the population until a suitable solution is found.

Four mutation operators were developed:

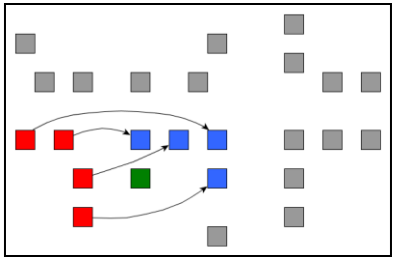
Single-Object Replacement

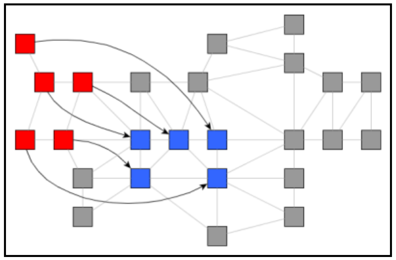




Two-Seed Set Reconstruction

One-Seed Set Reconstruction





VF2 Subgraph Isomorphism

Single-Object Replacement (SOR)

Given a parent: $\Gamma_P = \left(x_{(1)}, x_{(2)}, \ldots, x_{(n)}\right)$

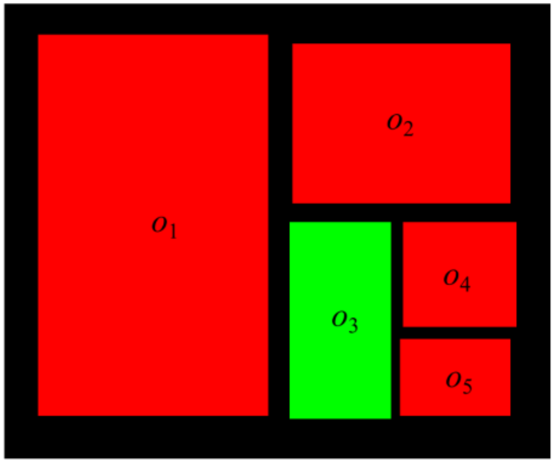
1. Pick an object from Γ_P to replace
2. Get the set of connected neighbors which could replace the selected object and still maintain full connectivity
3. For each neighbor
   a. Create a temporary child where the seed object is replaced by the neighbor and evaluate the fitness
4. Return the best child

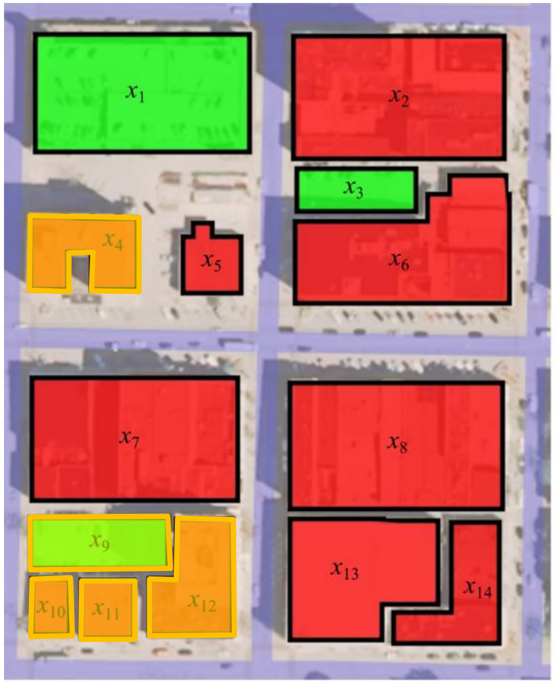
We typically run this algorithm several times with different parent permutations, using all possible replacement objects.

The complexity is then $O(pn^3\omega K)$.

- p is the number of permutations
- n is the number of objects in the sketch
- ω is the number of bins in each force histogram
- K is the maximum number of nearest neighbor connections

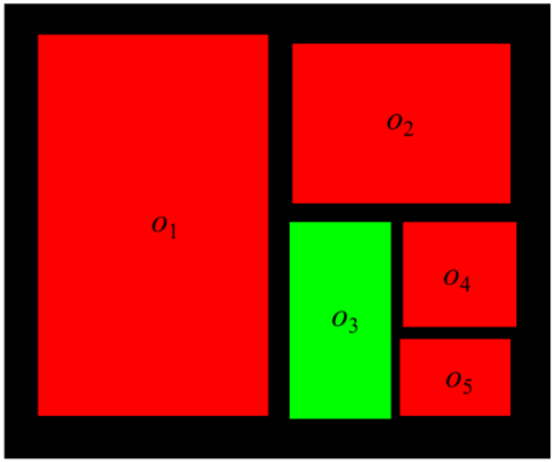
Single-Object Replacement (SOR)

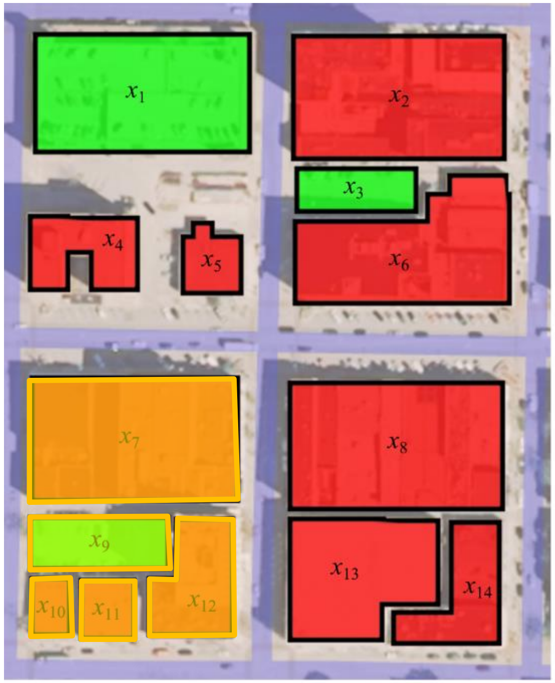




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_{12}, x_{10}, x_9, x_4, x_{11})$
Permute	$\Gamma' = (x_4, x_{12}, x_9, x_{11}, x_{10})$
Pick a single object	$c_i = x_4$
Get list of possible replacements	$\mathcal{X} = \{x_2, x_5, x_6, x_7, x_8, x_{13}, x_{14}\}$
Find best replacement	$x_{best} = x_7$
Replace c_i with x_{best}	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

Single-Object Replacement (SOR)





Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_{12}, x_{10}, x_9, x_4, x_{11})$
Permute	$\Gamma' = (x_4, x_{12}, x_9, x_{11}, x_{10})$
Pick a single object	$c_i = x_4$
Get list of possible replacements	$\mathcal{X} = \{x_2, x_5, x_6, x_7, x_8, x_{13}, x_{14}\}$
Find best replacement	$x_{best} = x_7$
Replace c_i with x_{best}	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

One-Seed Set Reconstruction

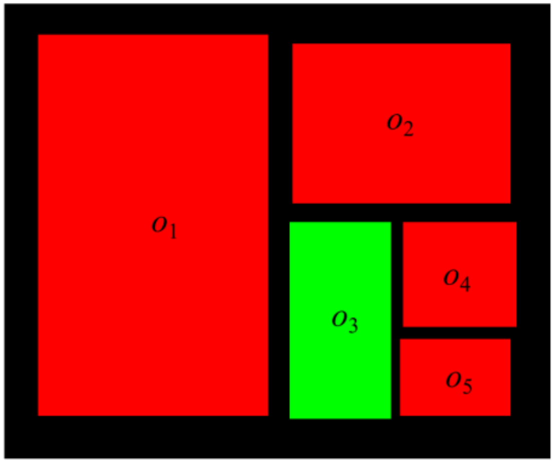
Given a parent: $\Gamma_P = \left(x_{(1)}, x_{(2)}, \dots, x_{(n)}\right)$

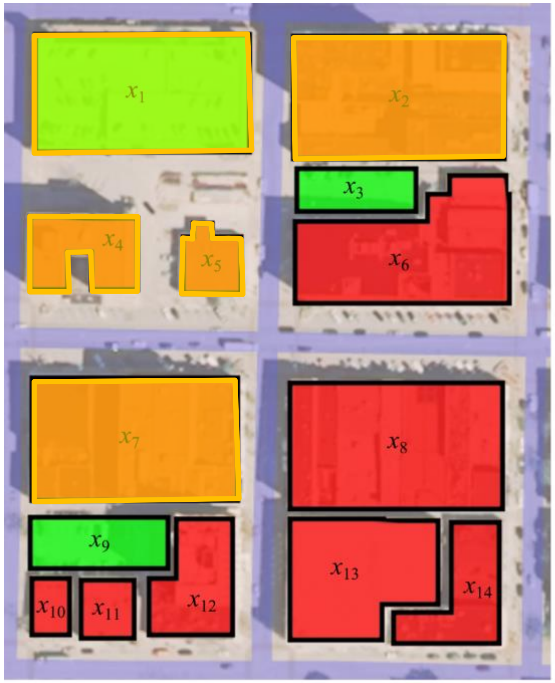
1. Pick an object from Γ_P as the seed object and random order for gene replacement, σ
2. Create an empty solution Γ' and assign the seed object to $\Gamma'(\sigma_1)$
3. Until Γ' is a complete solution
   a. Get the nearest neighbors of the current solution, Γ'
   b. Pick the best neighbor for the next gene location using the same method as SOR mutation
   c. Add this neighbor to Γ' and move to the next gene location
4. Return Γ' as the improved child

Usually this algorithm is run with all possible seed objects and all possible initial seed locations.

The complexity is $O(n^5 \omega K)$.

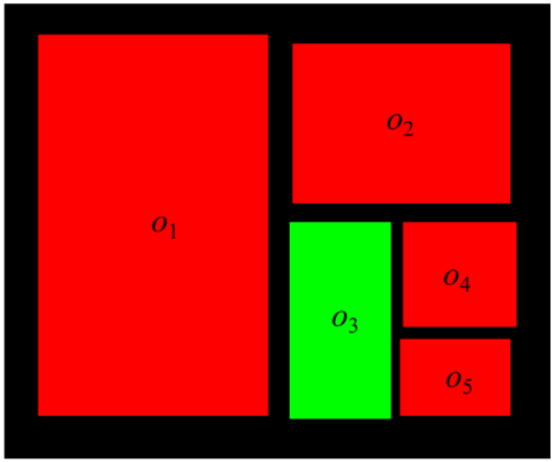
One-Seed Set Reconstruction

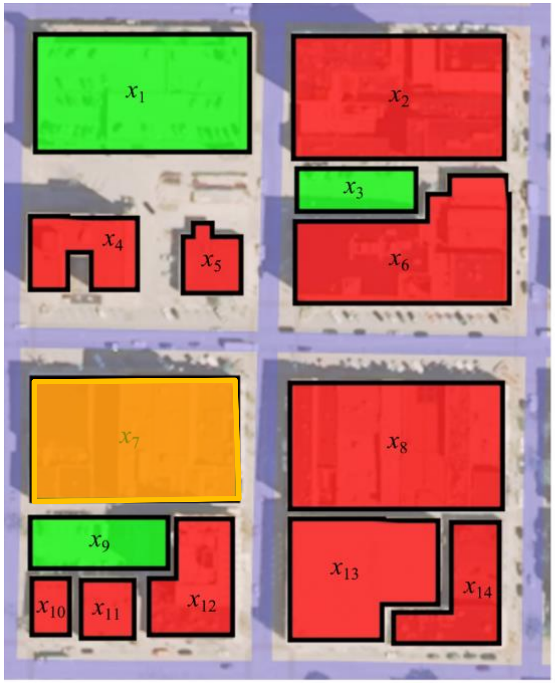




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_2, x_5, x_1, x_4, x_7)$
Pick the seed object	$\Gamma'(o_1) = x_7$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_4$
Find best match for this object	$x_{best} = x_{11}$
Update chromosome	$\Gamma'(o_4) = x_{11}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

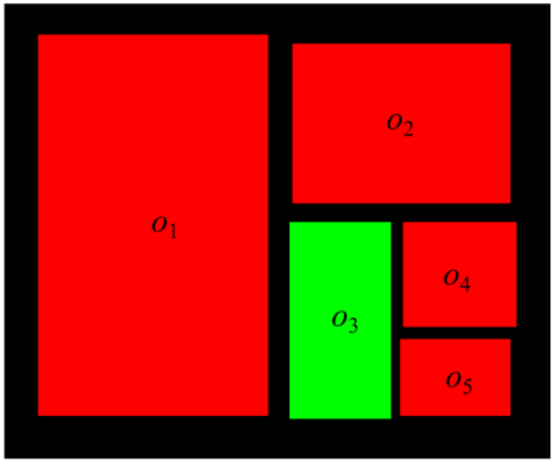
One-Seed Set Reconstruction

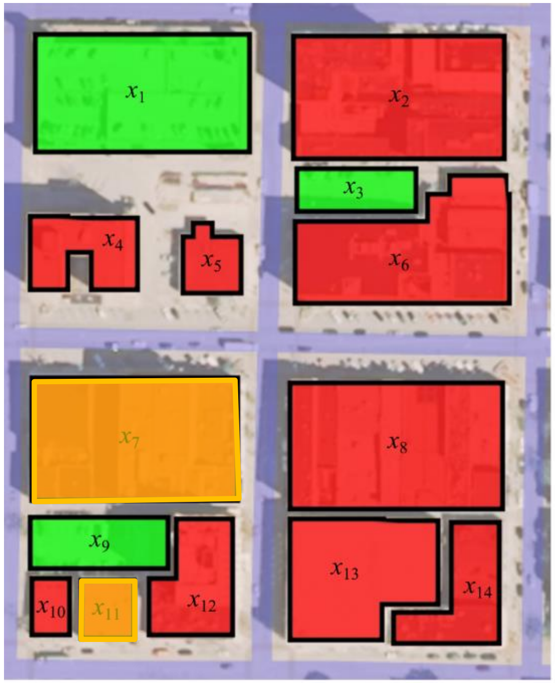




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_2, x_5, x_1, x_4, x_7)$
Pick the seed object	$\Gamma'(o_1) = x_7$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_4$
Find best match for this object	$x_{best} = x_{11}$
Update chromosome	$\Gamma'(o_4) = x_{11}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

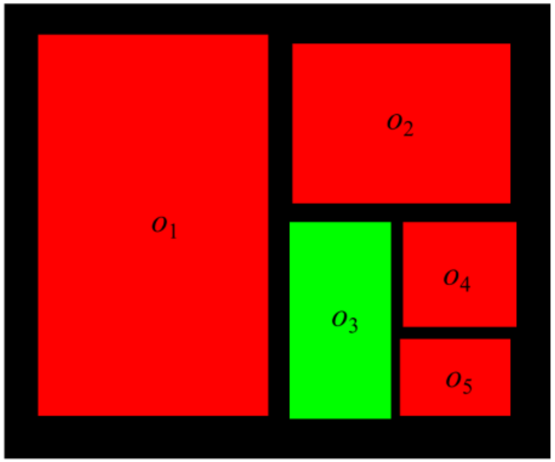
One-Seed Set Reconstruction

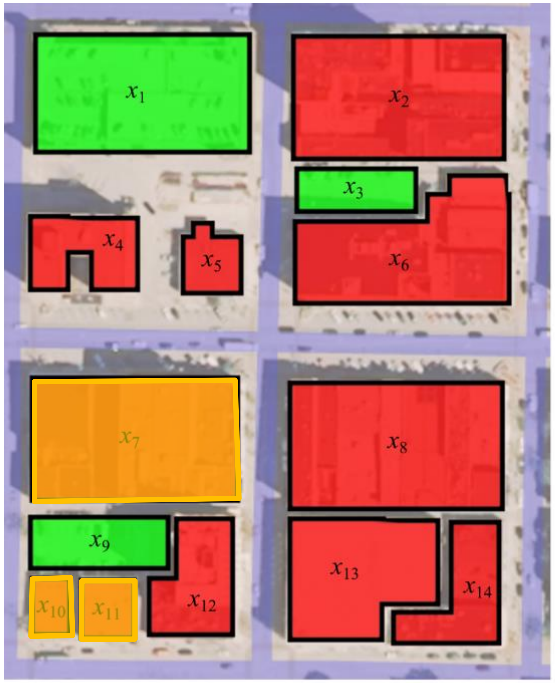




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_2, x_5, x_1, x_4, x_7)$
Pick the seed object	$\Gamma'(o_1) = x_7$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_4$
Find best match for this object	$x_{best} = x_{11}$
Update chromosome	$\Gamma'(o_4) = x_{11}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

One-Seed Set Reconstruction





Goal $\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$

Parent $\Gamma_P = (x_2, x_5, x_1, x_4, x_7)$

Pick the seed object $\Gamma'(o_1) = x_7$

Get list of neighbor objects $\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{14}\}$

Pick next sketch object randomly $o_k = o_4$

Find best match for this object $x_{best} = x_{11}$

Update chromosome $\Gamma'(o_4) = x_{11}$

Pick next sketch object randomly $o_k = o_5$

Find best match for this object $x_{best} = x_{10}$

Update chromosome $\Gamma'(o_5) = x_{10}$

Pick next sketch object randomly $o_k = o_3$

Find best match for this object $x_{best} = x_9$

Update chromosome $\Gamma'(o_3) = x_9$

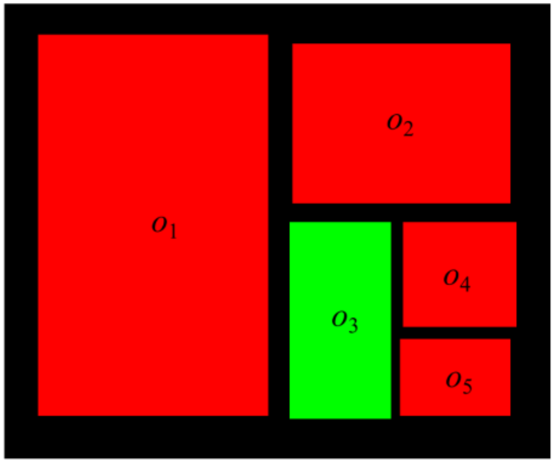
Pick next sketch object randomly $o_k = o_2$

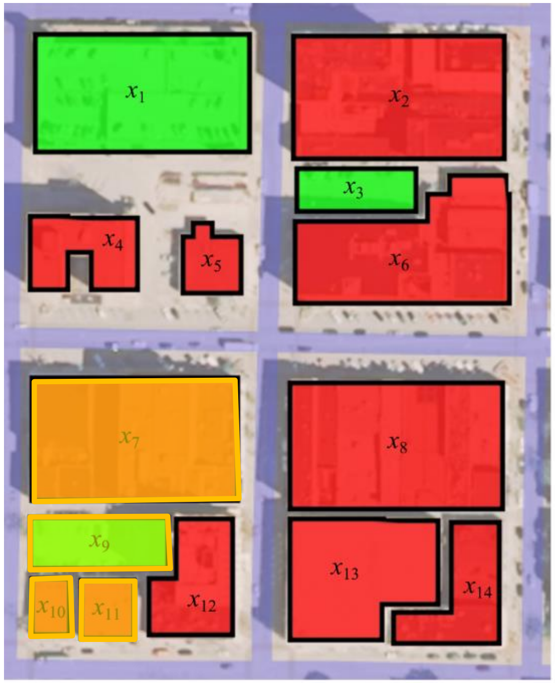
Find best match for this object $x_{best} = x_{12}$

Update chromosome $\Gamma'(o_2) = x_{12}$

Return child $\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

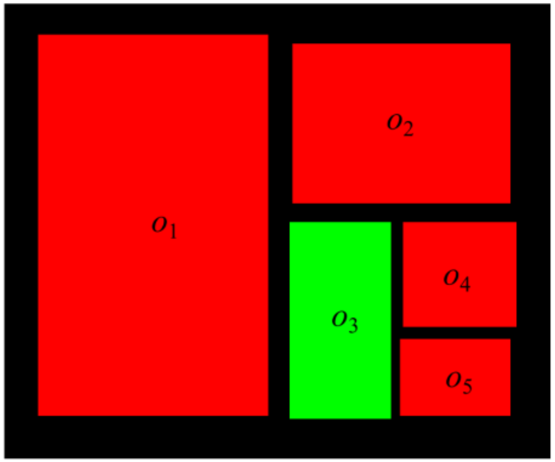
One-Seed Set Reconstruction

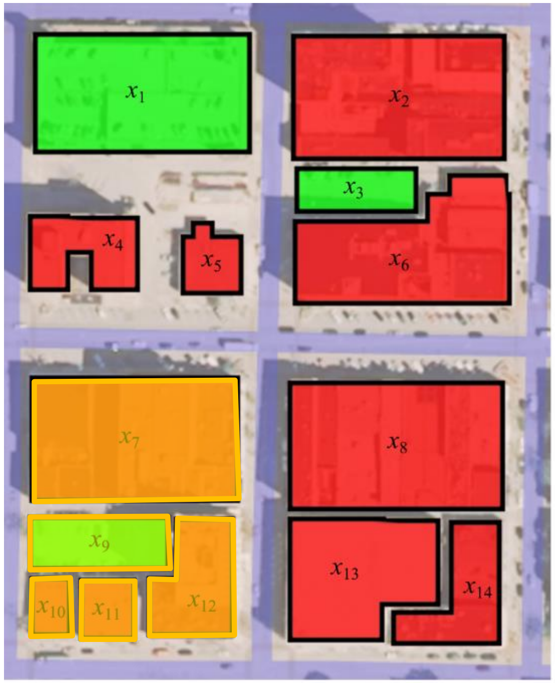




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_2, x_5, x_1, x_4, x_7)$
Pick the seed object	$\Gamma'(o_1) = x_7$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \ldots, x_6, x_8, \ldots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_4$
Find best match for this object	$x_{best} = x_{11}$
Update chromosome	$\Gamma'(o_4) = x_{11}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

One-Seed Set Reconstruction





Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_2, x_5, x_1, x_4, x_7)$
Pick the seed object	$\Gamma'(o_1) = x_7$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_4$
Find best match for this object	$x_{best} = x_{11}$
Update chromosome	$\Gamma'(o_4) = x_{11}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

Two-Seed Set Reconstruction

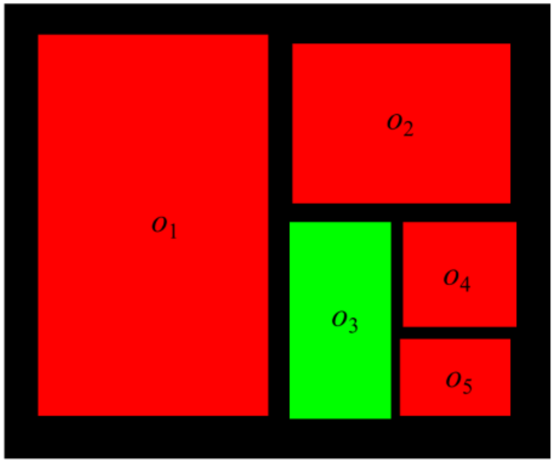
Given a parent: $\Gamma_P = \left(x_{(1)}, x_{(2)}, \dots, x_{(n)}\right)$

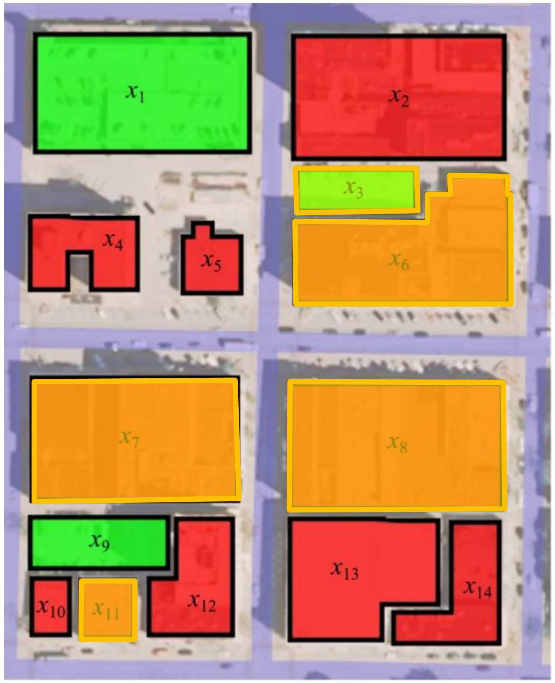
1. Pick two objects from Γ_P as the seed objects and a random order for gene replacement, σ
2. Create an empty solution Γ' and assign the seed objects to $\Gamma'(\sigma_1)$ and $\Gamma'(\sigma_2)$
3. Until Γ' is a complete solution
   a. Get the nearest neighbors of the current solution, Γ'
   b. Pick the best neighbor for the next gene location using the same method as SOR mutation
   c. Add this neighbor to Γ' and move to the next gene location
4. Return Γ' as the improved child

This algorithm is run with all possible pairs of seed objects and all possible initial seed locations.

The complexity is $O(n^7 \omega K)$.

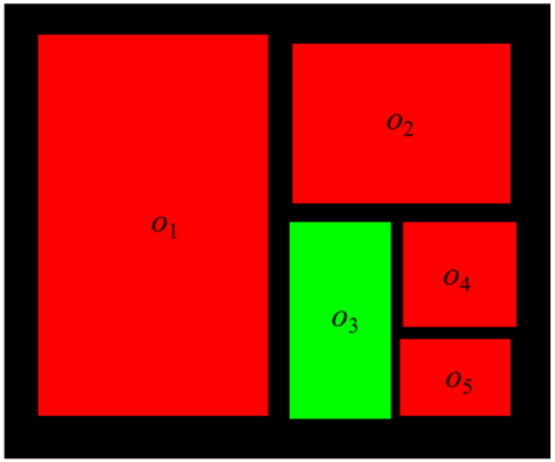
Two-Seed Set Reconstruction

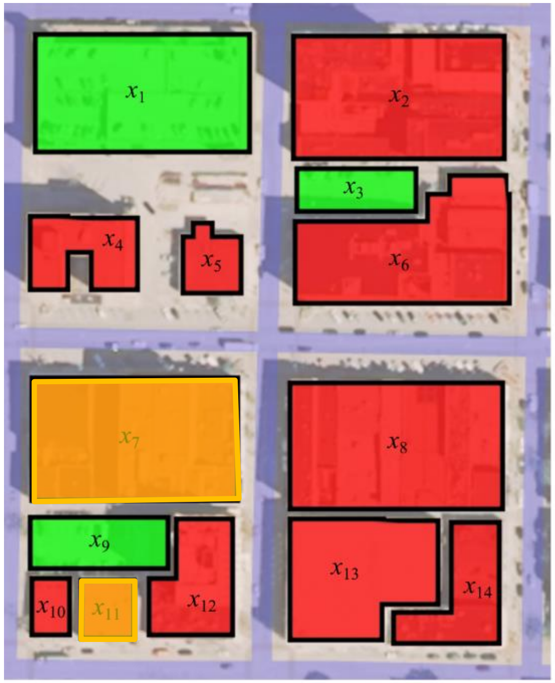




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_{11}, x_6, x_3, x_7, x_8)$
Pick the two seed objects	$\Gamma'(o_1) = x_7$ and $\Gamma'(o_4) = x_{11}$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{10}, x_{12}, \dots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

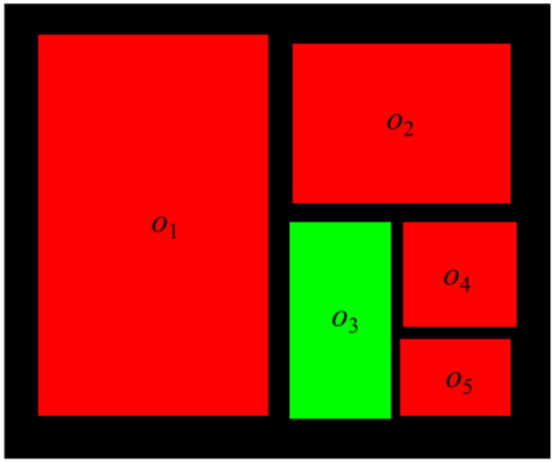
Two-Seed Set Reconstruction

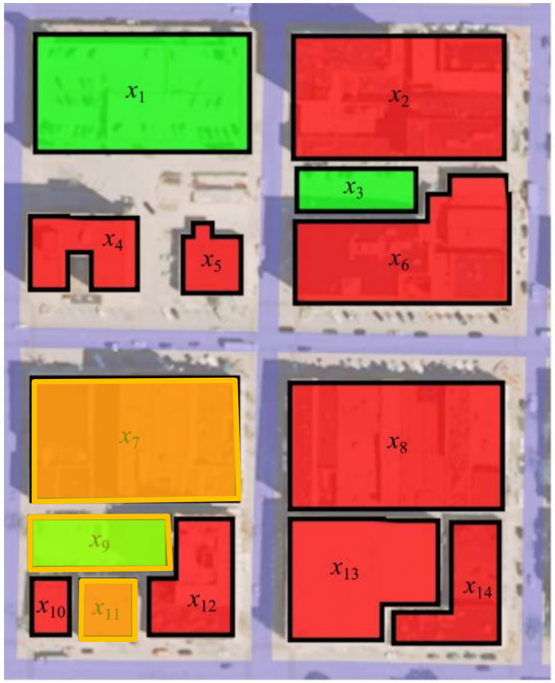




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_{11}, x_6, x_3, x_7, x_8)$
Pick the two seed objects	$\Gamma'(o_1) = x_7$ and $\Gamma'(o_4) = x_{11}$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{10}, x_{12}, \dots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

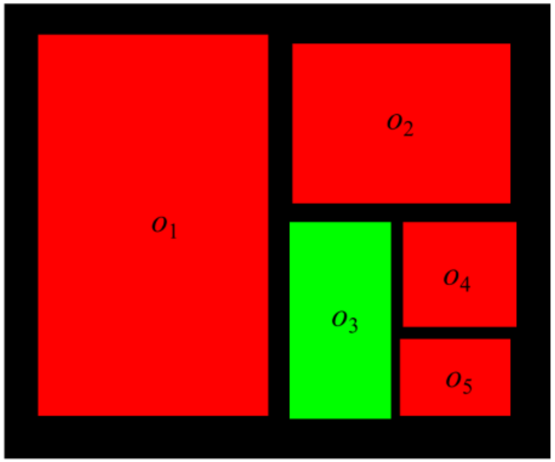
Two-Seed Set Reconstruction

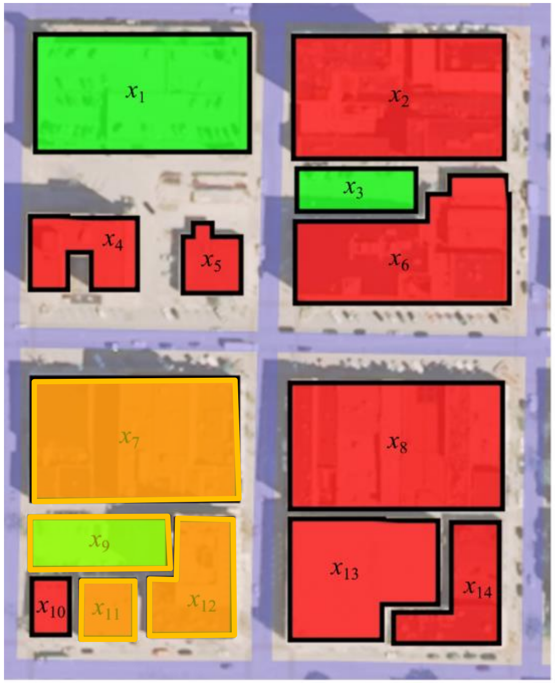




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_{11}, x_6, x_3, x_7, x_8)$
Pick the two seed objects	$\Gamma'(o_1) = x_7$ and $\Gamma'(o_4) = x_{11}$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{10}, x_{12}, \dots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

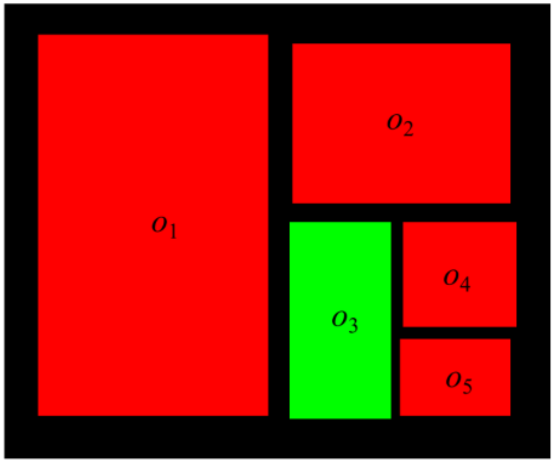
Two-Seed Set Reconstruction

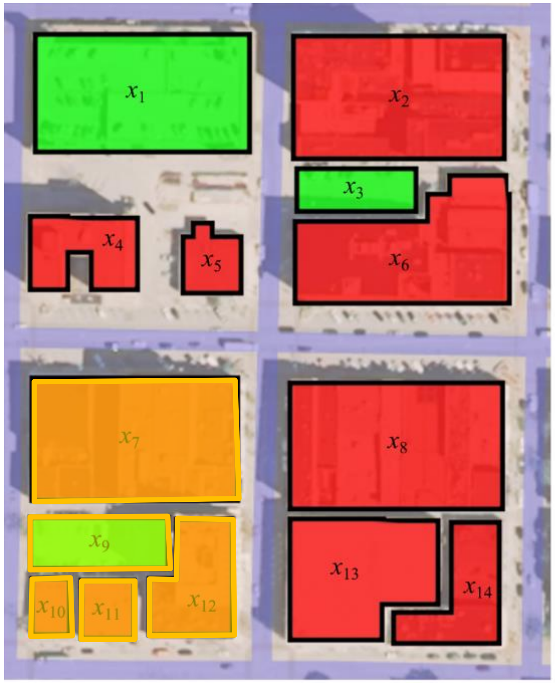




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_{11}, x_6, x_3, x_7, x_8)$
Pick the two seed objects	$\Gamma'(o_1) = x_7$ and $\Gamma'(o_4) = x_{11}$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \dots, x_6, x_8, \dots, x_{10}, x_{12}, \dots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

Two-Seed Set Reconstruction





Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_{11}, x_6, x_3, x_7, x_8)$
Pick the two seed objects	$\Gamma'(o_1) = x_7$ and $\Gamma'(o_4) = x_{11}$
Get list of neighbor objects	$\mathcal{X} = \{x_1, \ldots, x_6, x_8, \ldots, x_{10}, x_{12}, \ldots, x_{14}\}$
Pick next sketch object randomly	$o_k = o_3$
Find best match for this object	$x_{best} = x_9$
Update chromosome	$\Gamma'(o_3) = x_9$
Pick next sketch object randomly	$o_k = o_2$
Find best match for this object	$x_{best} = x_{12}$
Update chromosome	$\Gamma'(o_2) = x_{12}$
Pick next sketch object randomly	$o_k = o_5$
Find best match for this object	$x_{best} = x_{10}$
Update chromosome	$\Gamma'(o_5) = x_{10}$
Return child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10})$

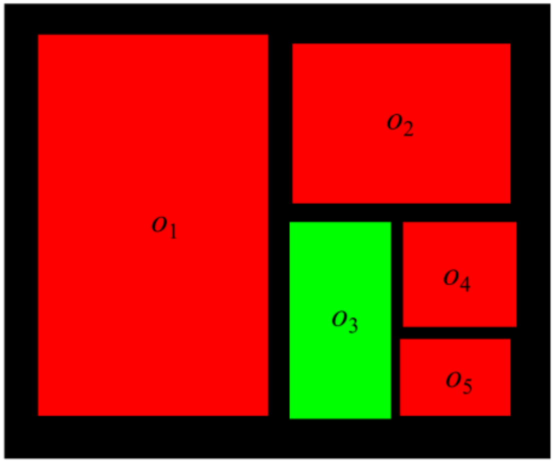
VF2 Subgraph Isomorphism

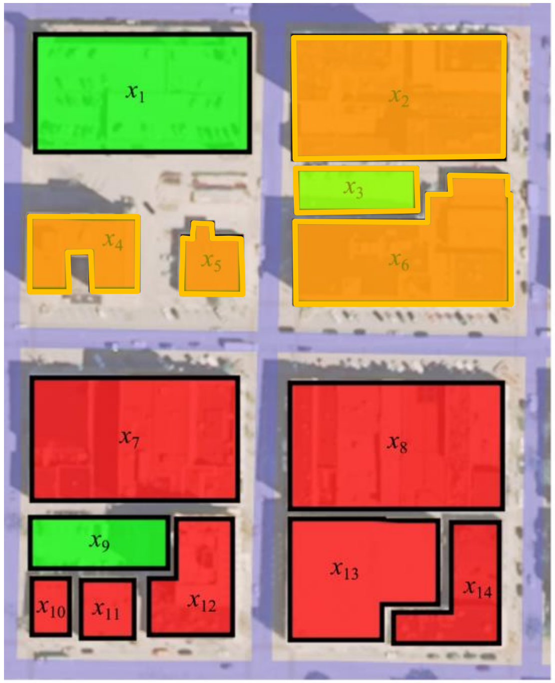
Given a parent: $\Gamma_P = \left(x_{(1)}, x_{(2)}, \ldots, x_{(n)}\right)$

1. Construct the graph G_S from the sketch and initialize the fitness threshold $\psi_{\min}$
2. Get the set of nearest neighbors $\mathcal{X}$ of the parent Γ_P
3. Shuffle the order of $\mathcal{X}$
4. Construct the nearest neighbor graph G_{NN} from $\mathcal{X}$
5. Run the VF2 algorithm on G_{NN} and G_S using the fitness threshold $\psi_{\min}$
6. If a match is found, return it as the improved child, otherwise decrease the value of $\psi_{\min}$ and go to 3

The time complexity of the VF2 algorithm ranges from $O(n\omega K)$ to $O(n!\,\omega K)$.

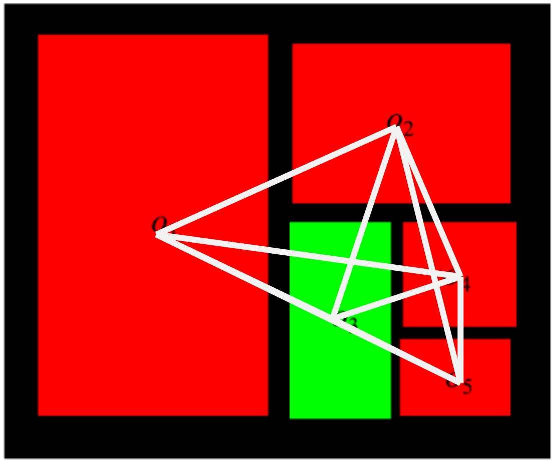
VF2 Subgraph Isomorphism

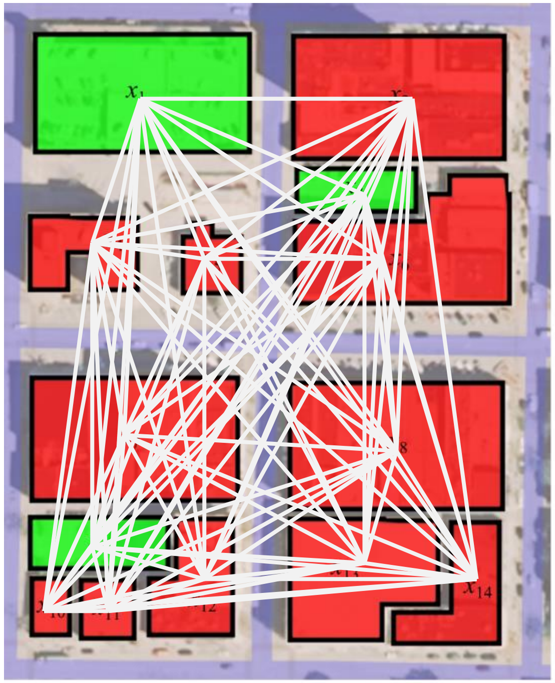




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_4, x_5, x_3, x_6, x_2)$
Build the sketch graph	$G_S = (\mathcal{O}, E_{\mathcal{O}}, L_{\mathcal{O}}, H_{\mathcal{O}})$
Get list of neighbor objects	$\mathcal{X} = \{x_1, x_2 \ldots, x_{14}\}$
Shuffle the order	$\mathcal{X} = \mathrm{shuffe}(\mathcal{X})$
Build the nearest neighbor graph	$G_{NN} = (\mathcal{X}, E_{\mathcal{X}}, L_{\mathcal{X}}, H_{\mathcal{X}})$
Run VF2 algorithm	$\mathcal{C} = \mathrm{match}(G_{NN}, G_S)$
Return best child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10}) \in \mathcal{C}$

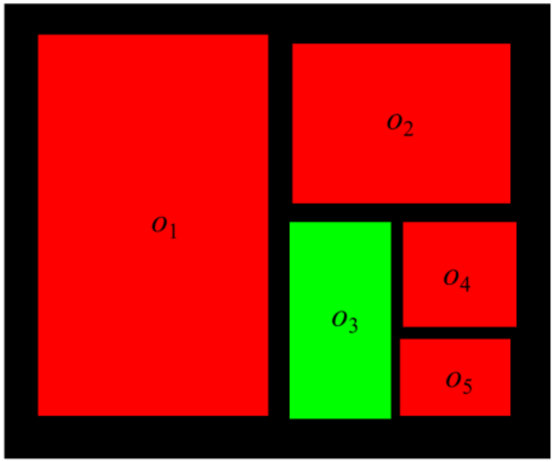
VF2 Subgraph Isomorphism

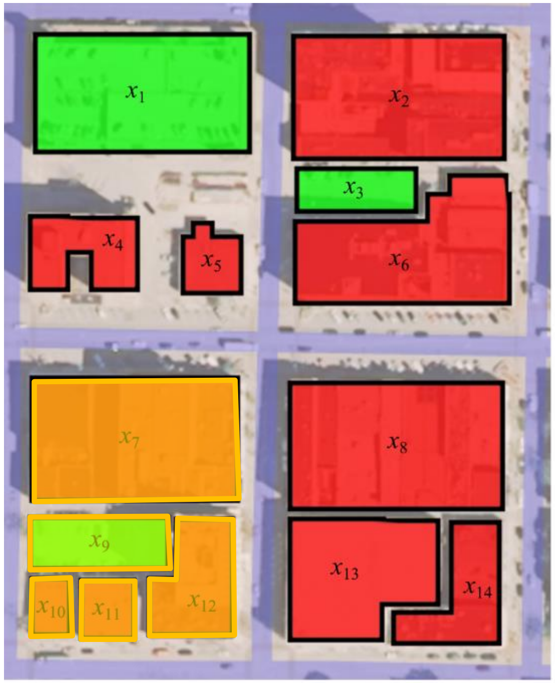




Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_4, x_5, x_3, x_6, x_2)$
Build the sketch graph	$G_S = (\mathcal{O}, E_{\mathcal{O}}, L_{\mathcal{O}}, H_{\mathcal{O}})$
Get list of neighbor objects	$\mathcal{X} = \{x_1, x_2 \ldots, x_{14}\}$
Shuffle the order	$\mathcal{X} = \mathrm{shuffe}(\mathcal{X})$
Build the nearest neighbor graph	$G_{NN} = (\mathcal{X}, E_{\mathcal{X}}, L_{\mathcal{X}}, H_{\mathcal{X}})$
Run VF2 algorithm	$\mathcal{C} = \mathrm{match}(G_{NN}, G_S)$
Return best child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10}) \in \mathcal{C}$

VF2 Subgraph Isomorphism



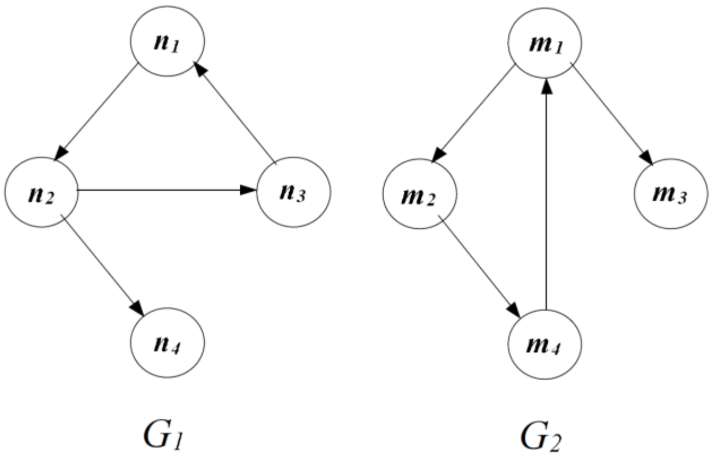


Goal	$\Gamma = (x_7, x_{12}, x_9, x_{11}, x_{10})$
Parent	$\Gamma_P = (x_4, x_5, x_3, x_6, x_2)$
Build the sketch graph	$G_S = (\mathcal{O}, E_{\mathcal{O}}, L_{\mathcal{O}}, H_{\mathcal{O}})$
Get list of neighbor objects	$\mathcal{X} = \{x_1, x_2 \ldots, x_{14}\}$
Shuffle the order	$\mathcal{X} = \text{shuffe}(\mathcal{X})$
Build the nearest neighbor graph	$G_{NN} = (\mathcal{X}, E_{\mathcal{X}}, L_{\mathcal{X}}, H_{\mathcal{X}})$
Run VF2 algorithm	$\mathcal{C} = \text{match}(G_{NN}, G_S)$
Return best child	$\Gamma' = (x_7, x_{12}, x_9, x_{11}, x_{10}) \in \mathcal{C}$

VF2 Algorithm

Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ find a mapping $M = \{(n, m) \in V_1 \times V_2 | n \text{ is mapped onto } m\}$.

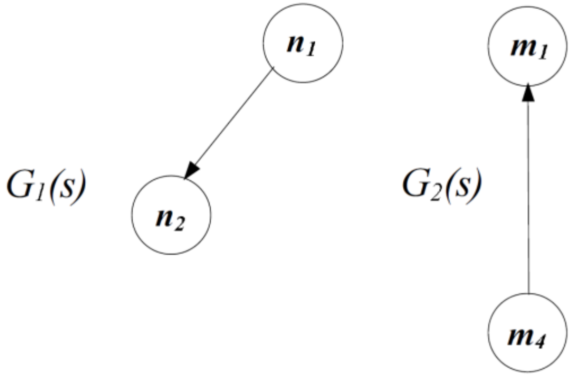
Use a state space representation in which each state is a partial mapping $M(s)$.



$M = \{(n_1,m_4), (n_2,m_1), (n_3,m_2), (n_4,m_3)\}$

$M(s) = \{(n_1,m_4), (n_2,m_1)\}$

$M_1(s) = \{n_1,n_2\}$ $M_2(s) = \{m_4,m_1\}$



[Cordella, et al., 1999]

VF2 Algorithm

VF2 Graph Matching Algorithm

```
Procedure: $\text{Match}(G_1, G_2, s)$
Input:   Graphs $G_1$ and $G_2$
         Intermediate state $s$; the initial state $s_0$ has $M(s_0) = \emptyset$

If $M(s)$ covers all the nodes of $G_2$ Then
    Output: $M(s)$
Else
    Compute the set $P(s)$ of the pairs candidate for inclusion in $M(s)$
    For Each $p = (n, m)$ in $P(s)$
        If $F(s, n, m) == \text{TRUE}$ Then
            Create a new state $s'$ by adding $p$ to $M(s)$
            Call $\text{Match}(G_1, G_2, s')$
        End If
    End For
End If
```

VF2 Feasibility Function

The key to the algorithm is the feasibility function that determines whether or not two nodes can be added to the partial map.

$$F(s,n,m) = F_{syn}(s,n,m) \land F_{sem}(s,n,m)$$

$F_{syn}(s,n,m)$ evaluates the syntactic structure of the graphs.

$F_{sem}(s,n,m)$ evaluates the semantic compatibility.

We define the semantic feasibility function as TRUE if the fitness of the partial map is greater than the threshold $\psi_{\min}$.

VF2 Syntactic Feasibility Rules

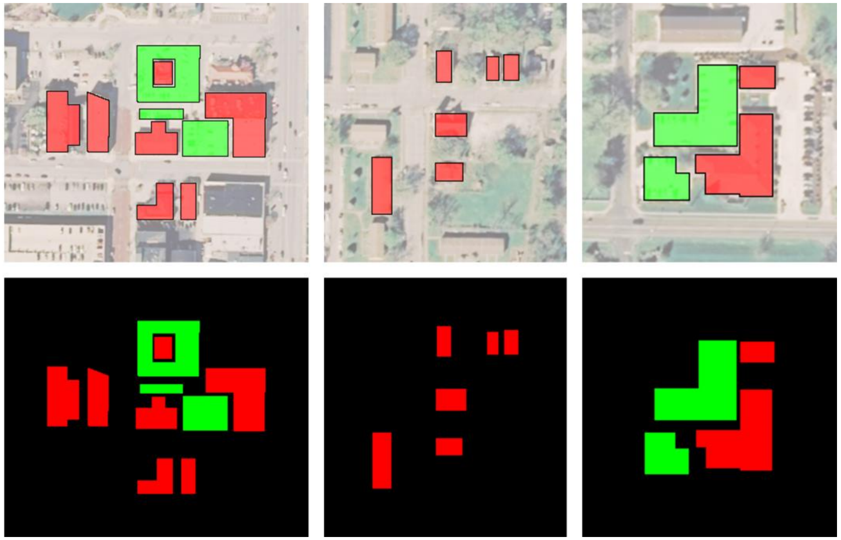
$$F_{syn}(s,n,m) = R_{\text{pred}} \wedge R_{\text{succ}} \wedge R_{\text{in}} \wedge R_{\text{out}} \wedge R_{\text{new}}$$

Look-Ahead	Rule	Condition
0	R_{pred}	Iff for each predecessor n' of n in the partial mapping, the corresponding node m' is a predecessor of m, and vice versa.
	R_{succ}	Iff for each successor n' of n in the partial mapping, the corresponding node m' is a successor of m, and vice versa.
1	R_{in}	Iff the number of predecessors (successors) of n that are in $T_1^{\text{in}}(s)$ is greater than or equal to the number of predecessors (successors) of m that are in $T_2^{\text{in}}(s)$.
	R_{out}	Iff the number of predecessors (successors) of n that are in $T_1^{\text{out}}(s)$ is greater than or equal to the number of predecessors (successors) of m that are in $T_2^{\text{out}}(s)$.
2	R_{new}	Iff the number of predecessors (successors) of n that are neither in $M_1(s)$ nor in $T_1(s)$ (new models) is greater than or equal to the number of predecessors (successors) of m that are neither in $M_2(s)$ nor in $T_2(s)$.

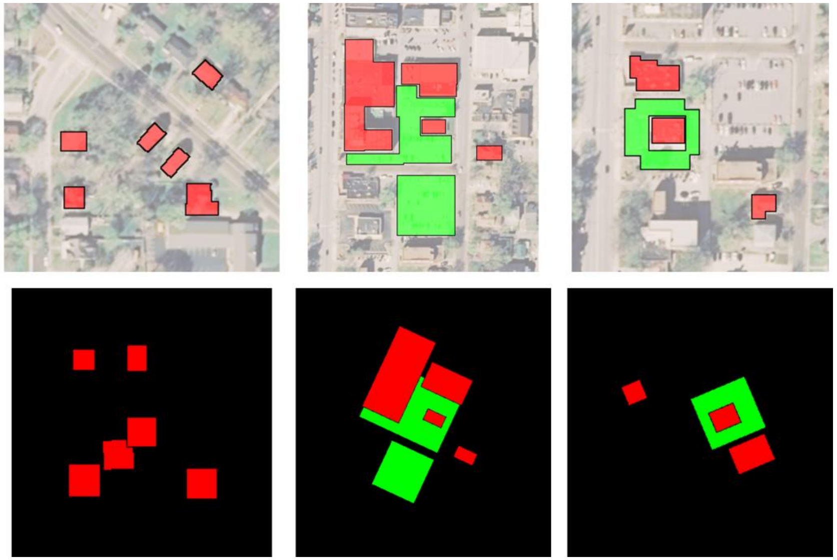
Experiments

We evaluate the performance of the matching algorithm using resubstitution sketches taken directly from the reference set, and simplified sketches, which have been reduced to bounding boxes and given an arbitrary orientation.

Resubstitution Sketches



Simplified Sketches



Mutation Method Comparison

To compare the mutation methods, 100 resubstitution sketches and 100 simplified sketches were randomly created, each containing 5 objects.

The search algorithm runs until the original sketch location is found with a maximum search time of 1000 generations. The results are averaged over 30 trials.

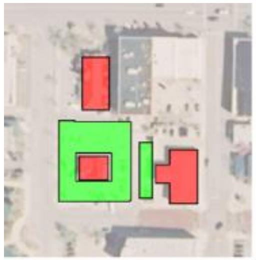
Mutation Method Comparison Search Parameters

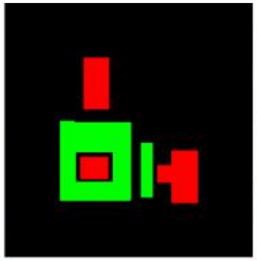
Mutation Method	Population Size (η)	Replacement Frequency (τ)	Replacement Percent (ρ)
SOR	50	50 Generations	50%
1-Seed	50	10 Generations	80%
2-Seed	50	10 Generations	80%
VF2	10	2 Generations	80%

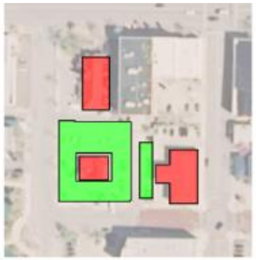
Mutation Method Comparison

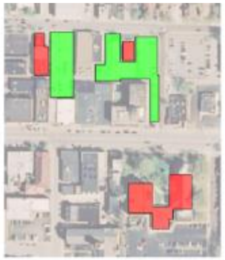
Results for Resubstitution Sketches

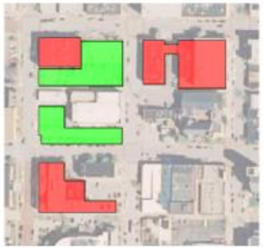
Mutation Method	Percent Found	Average Generations	Median Generations	Average Time (s)	Median Time (s)
SOR	51.9%	652	931	2594	2865
1-Seed	100%	14	5	159	44
2-Seed	99.4%	36	12	1703	494
VF2	100%	17	9	40	14

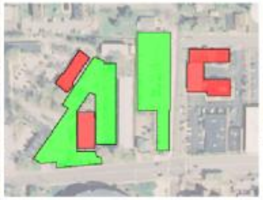

Ground Truth

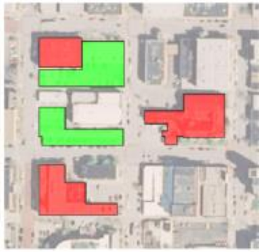

Sketch

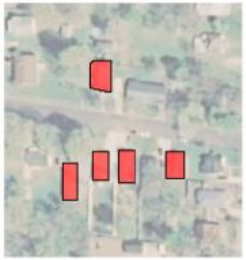

Fitness: 1.0

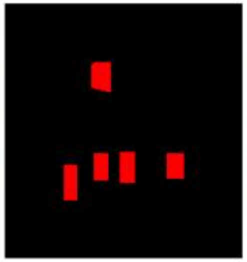

Fitness: 0.927

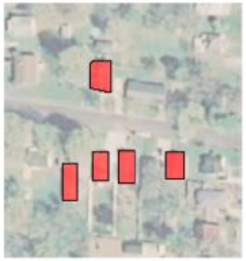

Fitness: 0.920

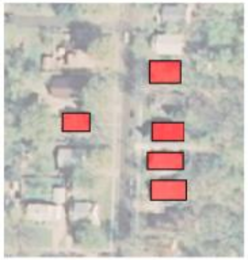

Fitness: 0.918

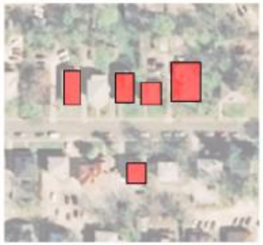

Fitness: 0.917

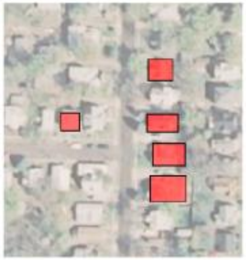

Ground Truth

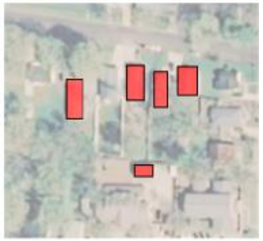

Sketch


Fitness: 1.0


Fitness: 0.994

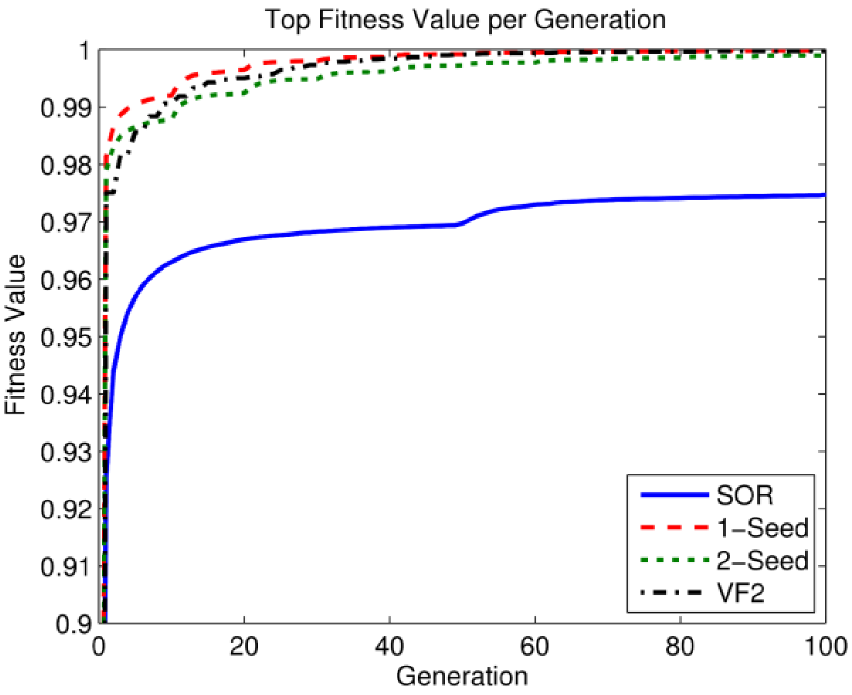

Fitness: 0.994

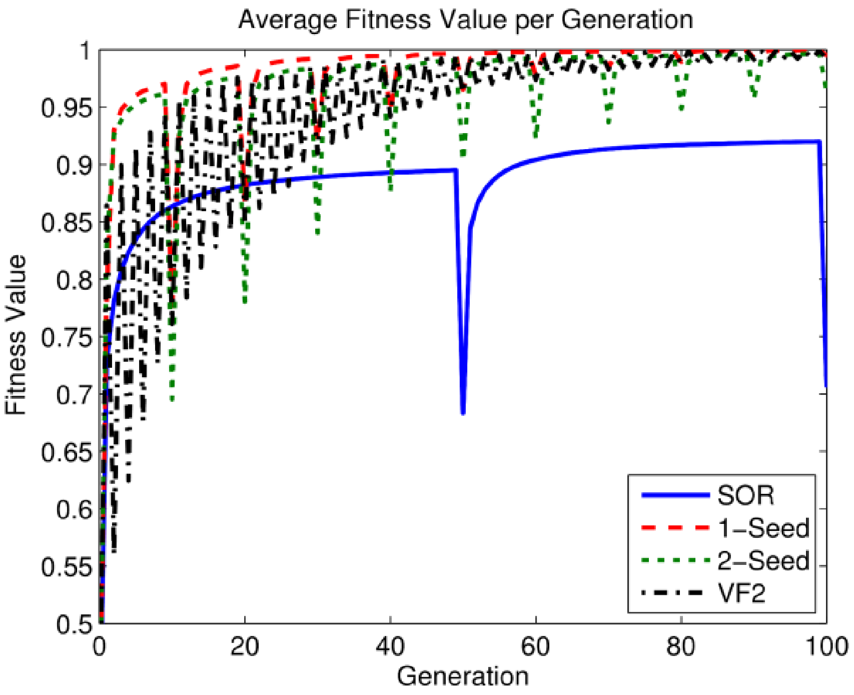

Fitness: 0.993


Fitness: 0.993

Mutation Method Comparison

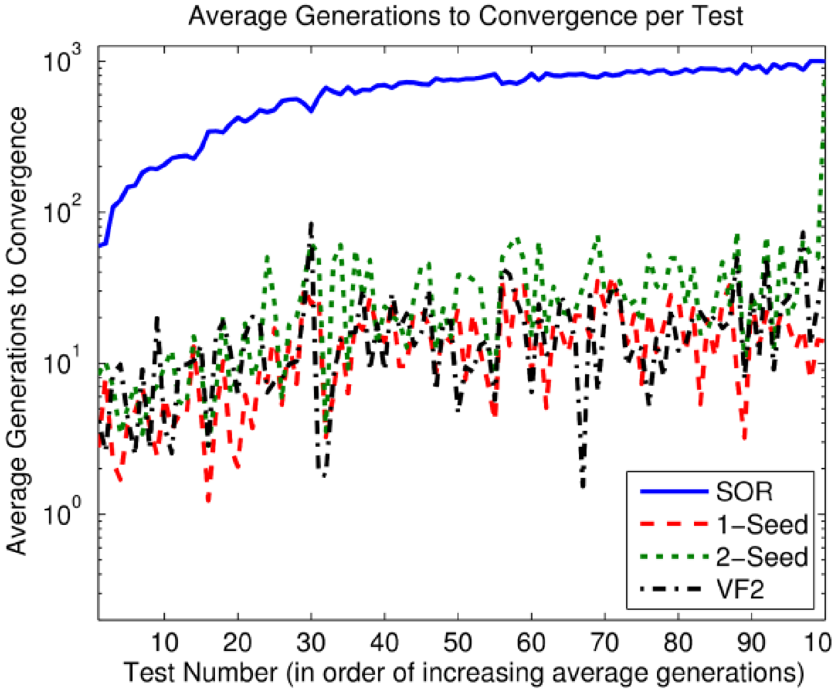
Results for Resubstitution Sketches

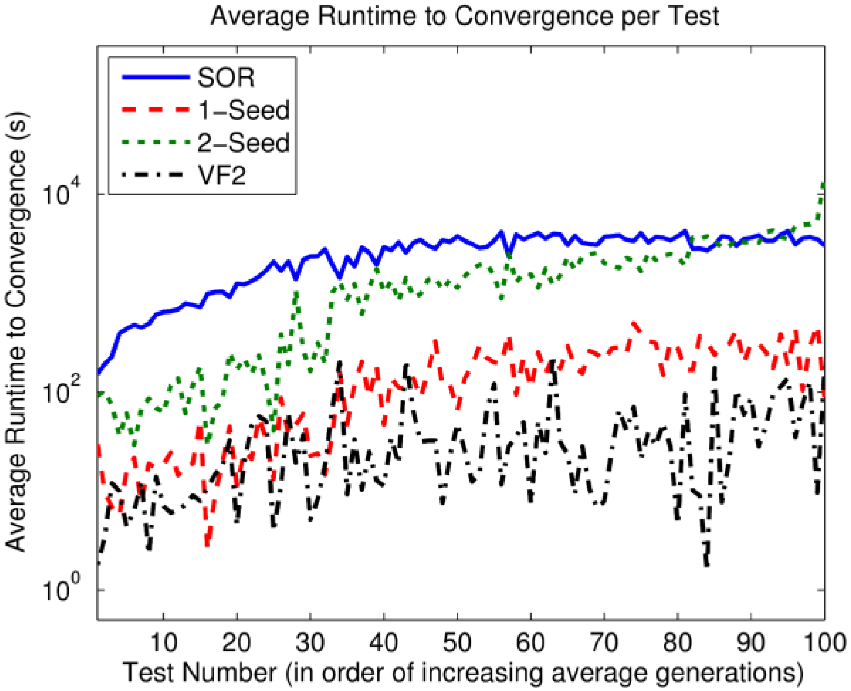




Mutation Method Comparison

Results for Resubstitution Sketches

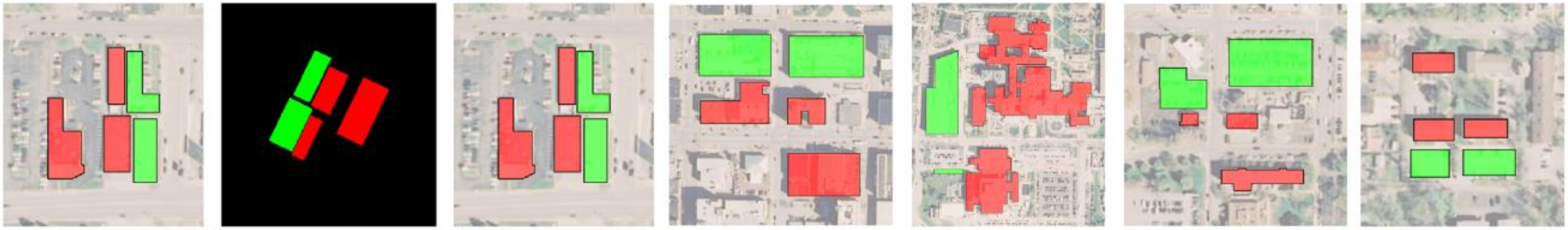




Mutation Method Comparison

Results for Simplified Sketches

Mutation Method	Percent Found	Average Generations	Median Generations	Average Time (s)	Median Time (s)
SOR	49.1%	681	1000	3127	3710
1-Seed	97.0%	60	12	539	103
2-Seed	96.5%	81	21	6404	1231
VF2	94.0%	81	11	216	19



Ground Truth

Sketch

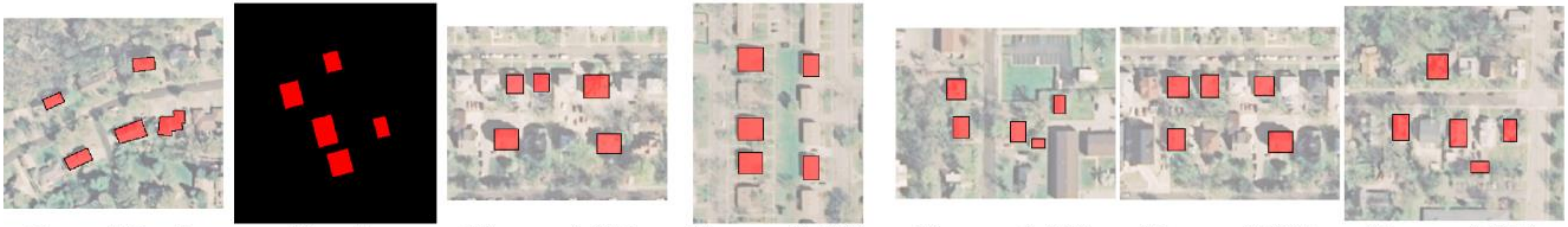
Fitness: 0.994

Fitness: 0.958

Fitness: 0.957

Fitness: 0.945

Fitness: 0.935



Ground Truth

Sketch

Fitness: 0.991

Fitness: 0.990

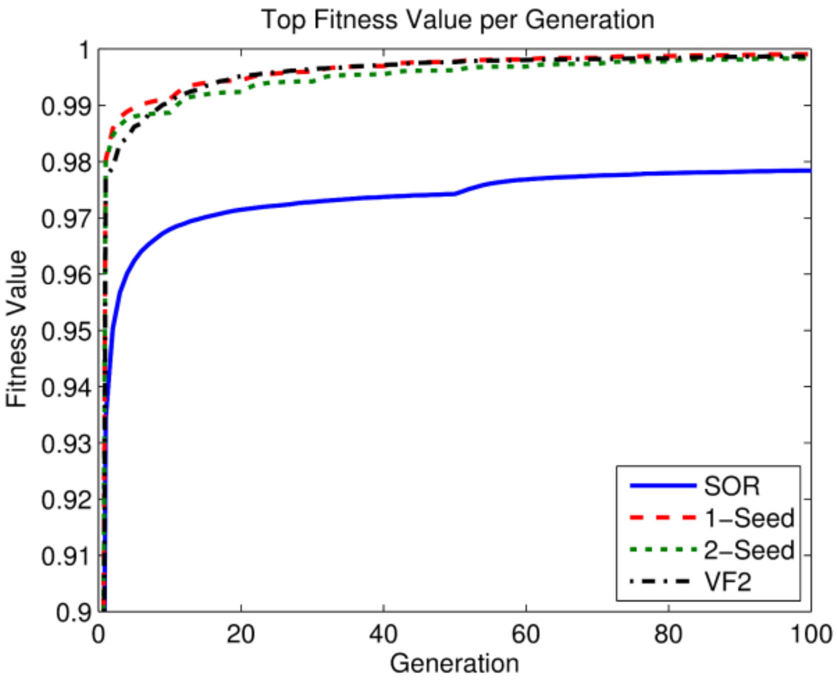
Fitness: 0.990

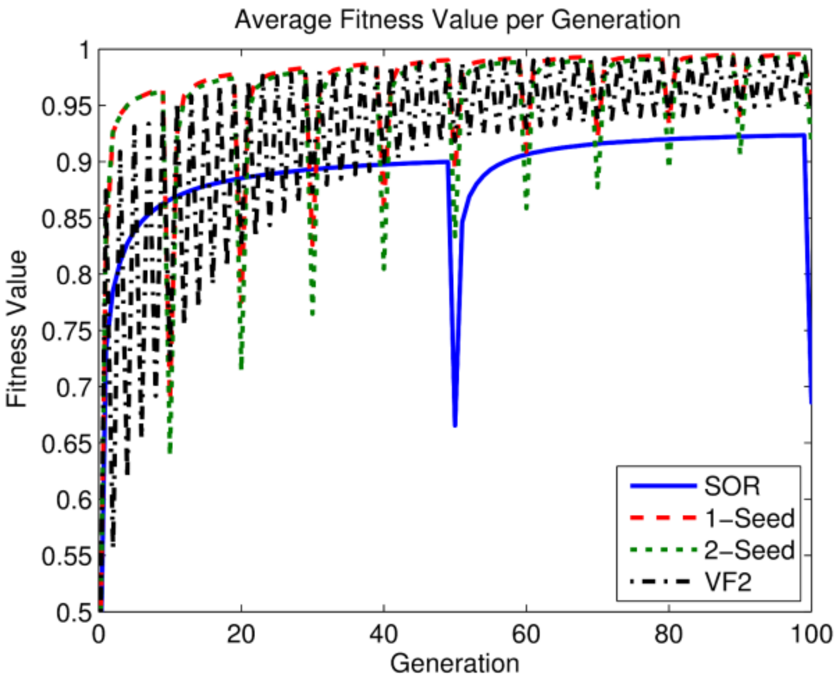
Fitness: 0.990

Fitness: 0.990

Mutation Method Comparison

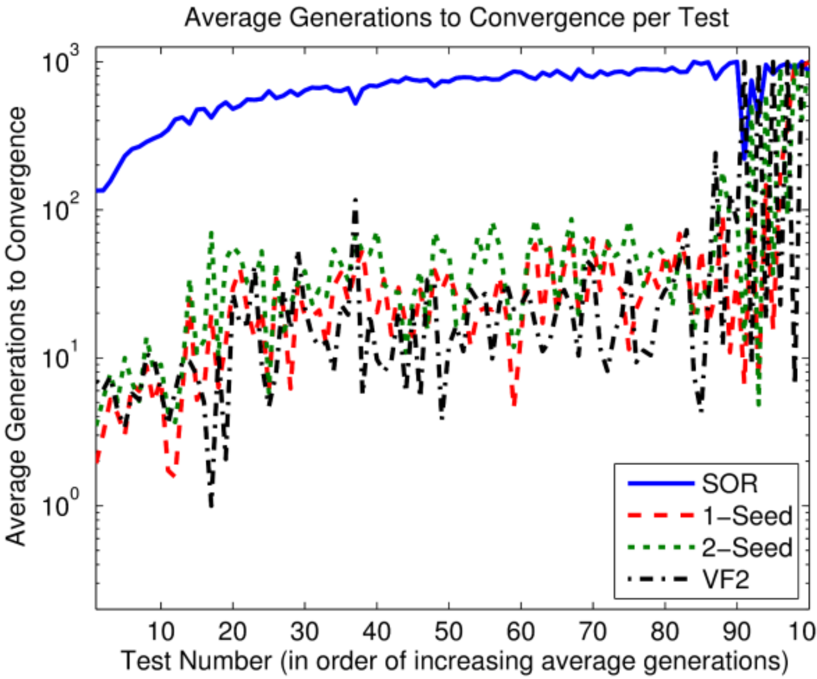
Results for Simplified Sketches

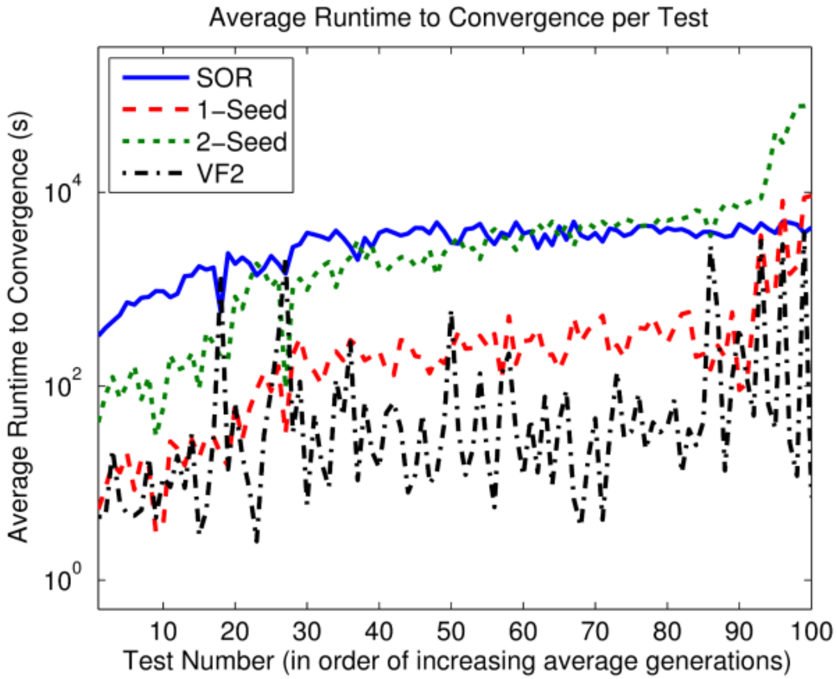




Mutation Method Comparison

Results for Simplified Sketches





Impact of Sketch Size

The first experiment suggested that the one-seed and the VF2 subgraph isomorphism mutation methods were the most promising.

To evaluate the impact of sketch size, 100 resubstitution sketches and 100 simplified sketches were randomly created, containing 4, 6, 8, 10, and 12 objects.

The search algorithm runs until the original sketch location is found with a maximum search time of 100 generations. The results are averaged over 10 trials.

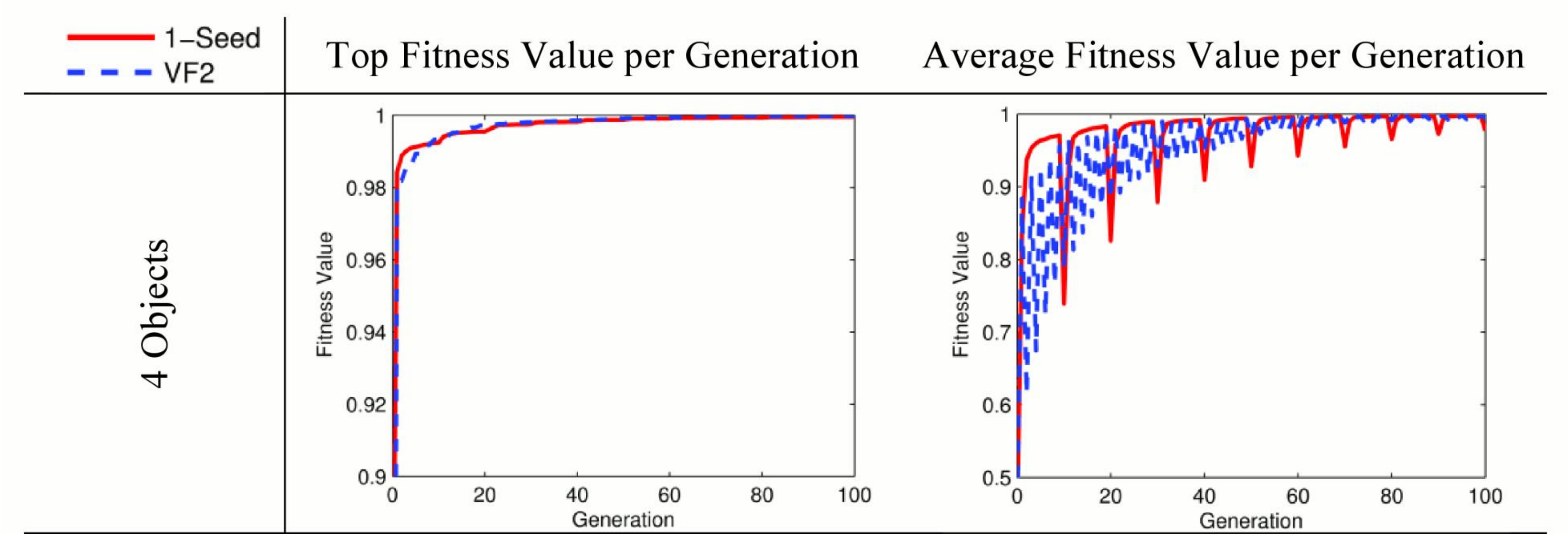
Impact of Sketch Size

Results for Resubstitution Sketches

Mutation Method	Number of Objects in Sketch	Percent Found	Average Generations	Median Generations	Average Time (s)	Median Time (s)
1-Seed	4	95.1%	22	11	89	40
	6	98.5%	12	3	207	40
	8	99.6%	9	2	494	83
	10	94.8%	13	2	1258	150
	12	86.2%	20	2	3304	502
VF2	4	98.7%	16	9	26	10
	6	96.6%	19	11	81	24
	8	98.1%	19	11	189	62
	10	90.8%	26	13	1777	563
	12	76.5%	39	23	4986	2617

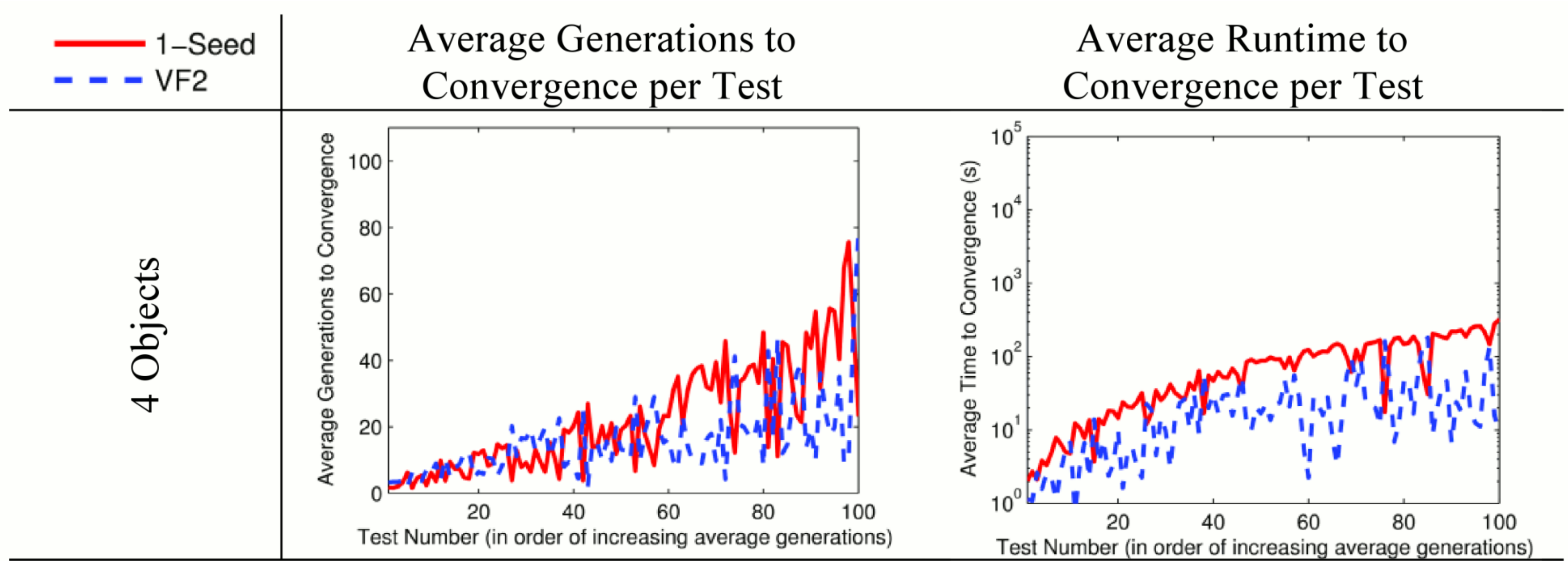
Impact of Sketch Size

Results for Resubstitution Sketches



Impact of Sketch Size

Results for Resubstitution Sketches



Average Generations to Convergence per Test

Average Runtime to Convergence per Test

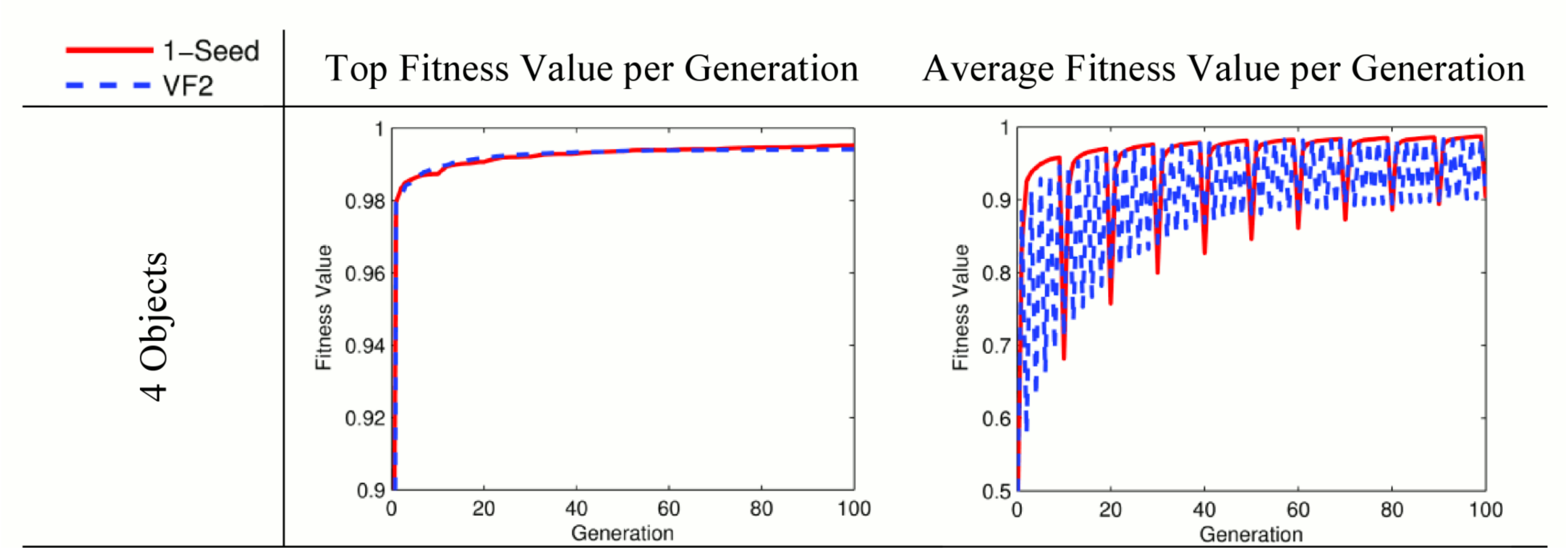
Impact of Sketch Size

Results for Simplified Sketches

Mutation Method	Number of Objects in Sketch	Percent Found	Average Generations	Median Generations	Average Time (s)	Median Time (s)
1-Seed	4	80.6%	36	18	151	60
	6	95.6%	16	5	318	84
	8	93.4%	18	5	771	194
	10	81.7%	28	7	2171	693
	12	87.2%	20	3	4317	874
VF2	4	80.4%	32	15	97	18
	6	94.3%	22	11	143	48
	8	86.0%	31	17	744	206
	10	69.2%	43	23	5927	1308
	12	78.8%	38	20	11009	3506

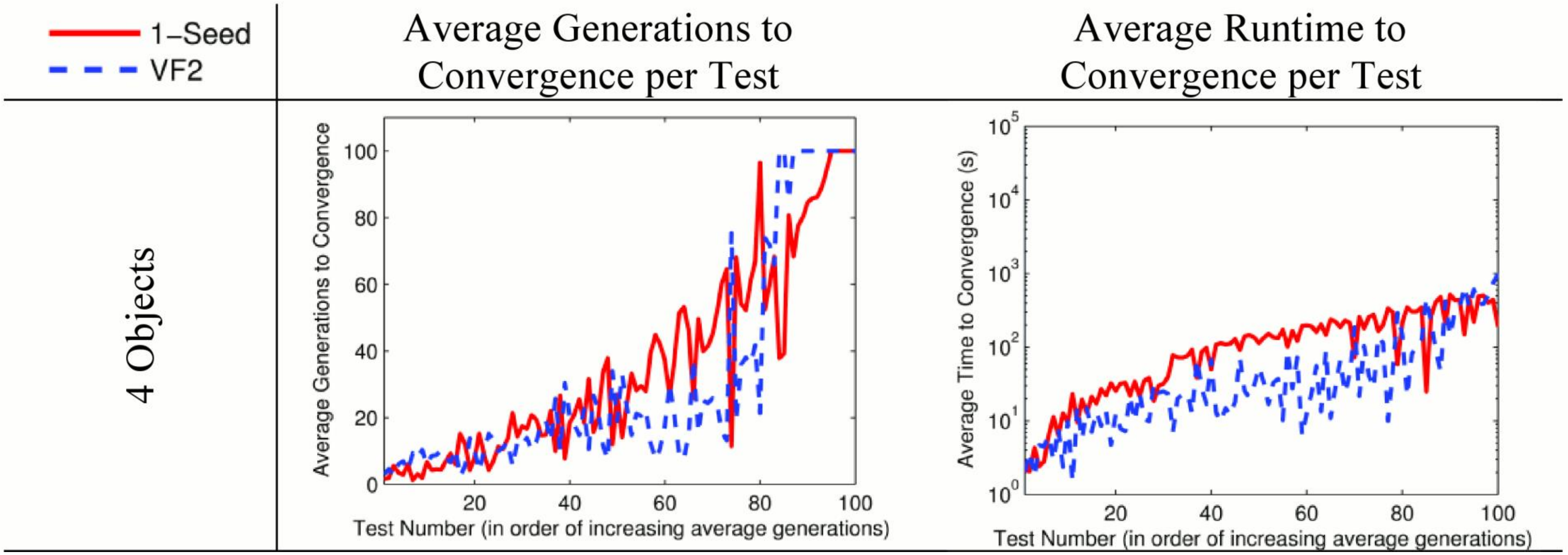
Impact of Sketch Size

Results for Simplified Sketches



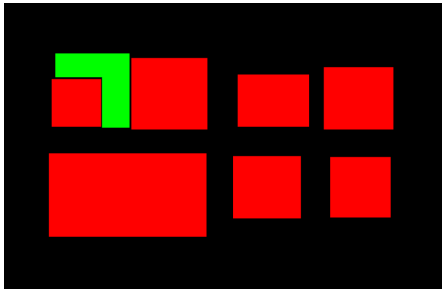
Impact of Sketch Size

Results for Simplified Sketches

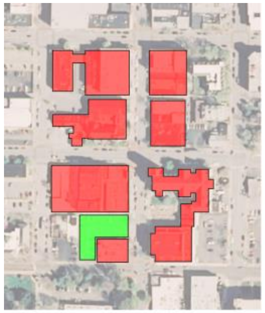


Real-World Example

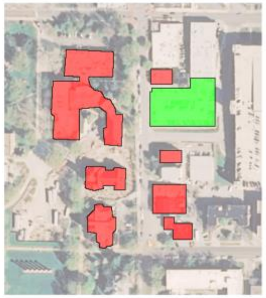
Hand-drafted Sketch



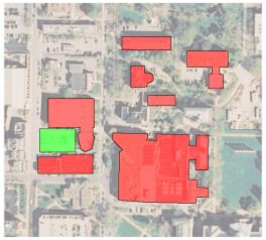
Top Results



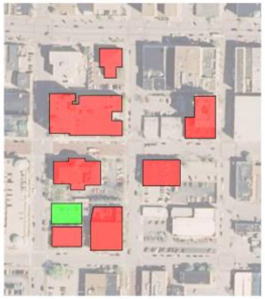
Fitness: 0.970



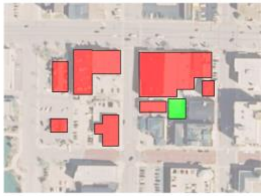
Fitness: 0.948



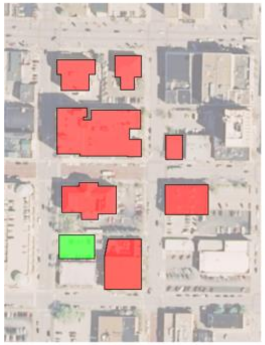
Fitness: 0.943



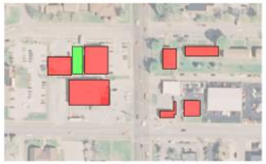
Fitness: 0.938



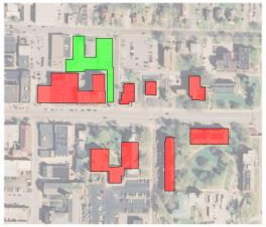
Fitness: 0.935



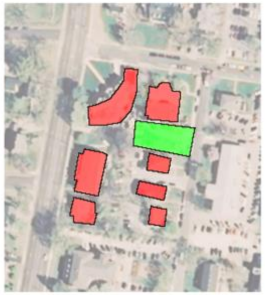
Fitness: 0.913



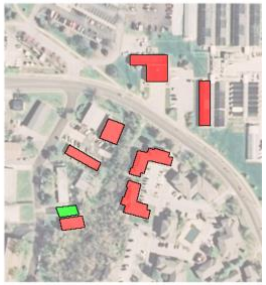
Fitness: 0.912



Fitness: 0.911



Fitness: 0.906



Fitness: 0.905

Conclusion

- The histograms of forces and ARG representation capture the relative spatial relationships between objects in a scene.

- This evolutionary framework allows a sketch to be searched for within a reference database.

- The one-seed and VF2 hybrid methods provide the best results.
  - VF2 works better for small sketches
  - One-seed is preferable for large sketches

References and Acknowledgement

[Cordella, et al., 1999] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "Performance evaluation of the VF graph matching algorithm," in *Proceedings of the International Conference on Image Analysis and Processing*, 1999, pp. 1172-1177.

[Matsakis & Wendling, 1999] P. Matsakis and L. Wendling, "A new way to represent the relative position between areal objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 21, pp. 634-643, 1999.

[Matsakis, et al., 2001] P. Matsakis, J. M. Keller, L. Wendling, J. Marjamaa, and O. Sjahputera, "Linguistic description of relative positions in images," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 31, pp. 573-588, 2001.

[Matsakis, et al., 2004] P. Matsakis, J. M. Keller, O. Sjahputera, and J. Marjamaa, "The use of force histograms for affine-invariant relative position description," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, pp. 1-18, 2004.

This work was funded by the U.S. National Geospatial-Intelligence Agency NURI grant HM 1582-08-1-0020.