

# Particle Filter Experiments

github: <https://github.com/helenlu66/ParticleFilter>

Helen Lu

March 18, 2024

## 1 Algorithm

Repeat steps 3-18 for each time step  $t$ : \_\_\_\_\_

---

### Algorithm 1 Particle Filter

---

- 1: **Initialization:**
  - 2: Draw a sample of  $N$  particles uniformly distributed in the map's 2D plane.
  - 3: Assign equal weights:  $w_0^{(i)} = \frac{1}{N}$ .
  - 4: **Move:**
  - 5: **for** each particle  $i$  from 1 to  $N$  **do**
  - 6:   move by applying the movement vector and add some randomness  $(x_t^{(i)}, y_t^{(i)}) = (x_{t-1}^{(i)}, y_{t-1}^{(i)}) + (dx, dy) + (N(0, \sigma^2), N(0, \sigma^2))$ .
  - 7: **end for**
  - 8: **Update Weights:**
  - 9: Observe the reference image at the true location
  - 10: Get a GPS reading of the true location
  - 11: **for** each particle  $i$  from 1 to  $N$  **do**
  - 12:   Observe the image at  $(x_t^{(i)}, y_t^{(i)})$ , calculate similarity to reference image using the Structural Similarity Index (SSIM)  $ssim(image^{(i)}, image_{ref})$
  - 13:   Calculate closeness (1.0 - normalized Euclidean distance away from GPS reading):  $close(state^{(i)} - state_{gps}) = 1.0 - normdist(state^{(i)} - state_{gps})$ .
  - 14:   Calculate a composite score for similarity:  $sim(obs, ref) = (1.0 - \gamma) * ssim(image^{(i)}, image_{ref}) + \gamma * close(state^{(i)} - state_{gps})$   $\triangleright \gamma$  is a hyperparameter we can tune
  - 15: **end for**
  - 16: **Roulette Resample:**
  - 17: Resample  $N * 95\%$  particles from the current set based on their weights to form a new set of particles.
  - 18: Add  $N * 5\%$  random particles to the resampled particles
  - 19: **Evaluate:**
  - 20: Calculate the Mean Squared Error of the top 10% nearest neighbors.
- 

The particle filter algorithm used for the experiments is described in the pseudocode 1. We used a composite score for estimating  $p(z|x)$ . We calculate the composite score by combining the similarity between the particle's image and the reference image with the closeness of the particle's location to the GPS reading of the true location:  $sim(obs, ref) = (1.0 - \gamma) * ssim(image^{(i)}, image_{ref}) + \gamma * close(state^{(i)} - state_{gps})$ . The similarity between the particle's image and the reference image is measured using the Structurl Similariy Index. We explore different values of  $\gamma$  in the experiments section.

## 1.1 Extra Credits Attempted

We incorporated noisy GPS readings (noise drawn from normal distribution centered at 0.0 with variance of 1.0), implemented a weighted distance-based and image observation based similarity measure, and experimented with white noise.

## 2 Experiments

The experiments explore changes in the following conditions:

- the observation size: 25, 50, 100
- the number of particles: 100, 500, 1000
- the weight used to produce the composite similarity score: 0.0, 0.2, 0.5, 0.8, 1.0
- the level of white noise added to the reference image: 0.0, 0.05, 0.1, 0.25, 0.5, 0.75, 1.0

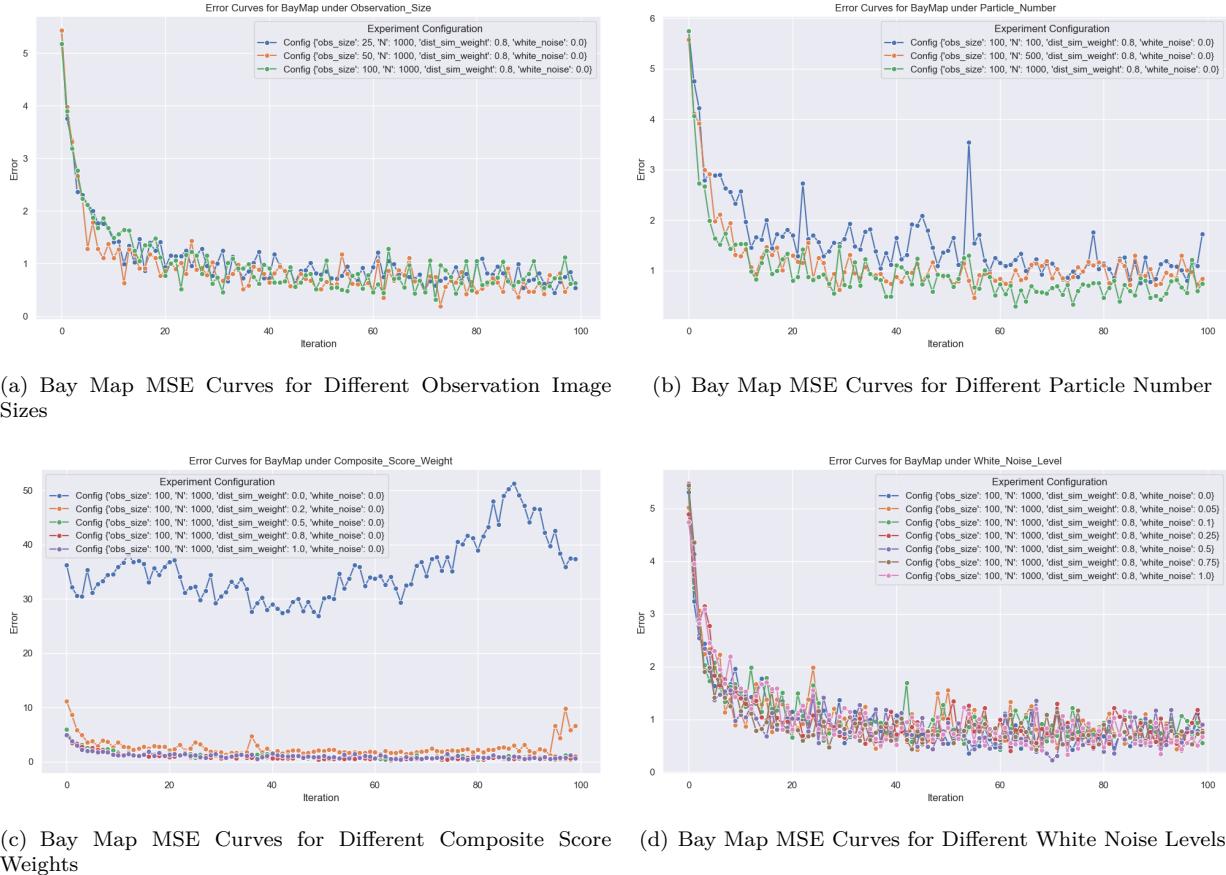


Figure 1: MSE Curves for Bay Map

The default configuration sets the observation image size to 100, the number of particles to 1000, the weight used in the composite score to 0.8, and the white noise level to 0.0. In each experiment condition, we vary the level of one variable. The other variables remain the same levels as that of the default configuration. We run 10 trials per level per condition and average across trials to get the error curves. The error is the mean squared error between the true location and the k nearest neighbors with the k being set to 10% of the number of particles. At each timestep, the filter gets a GPS location with added Gaussian noise centered around 0 with variance=1.0. We also circle the particles and the true locations on the map every 10 iterations to see if the particles are clustering around the true locations. Images generated from the experiments for each map are in the github repository in the directory named after the map.

### 3 Results

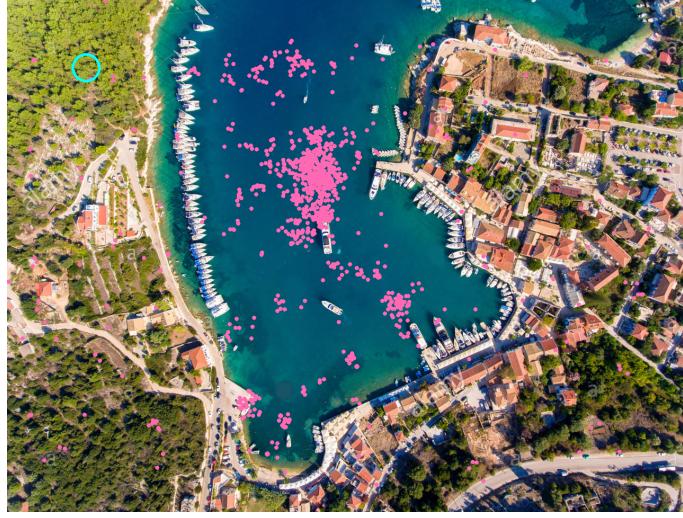


Figure 2: Bay Map Particles Clustering to the Wrong Location When  $\gamma=0.0$

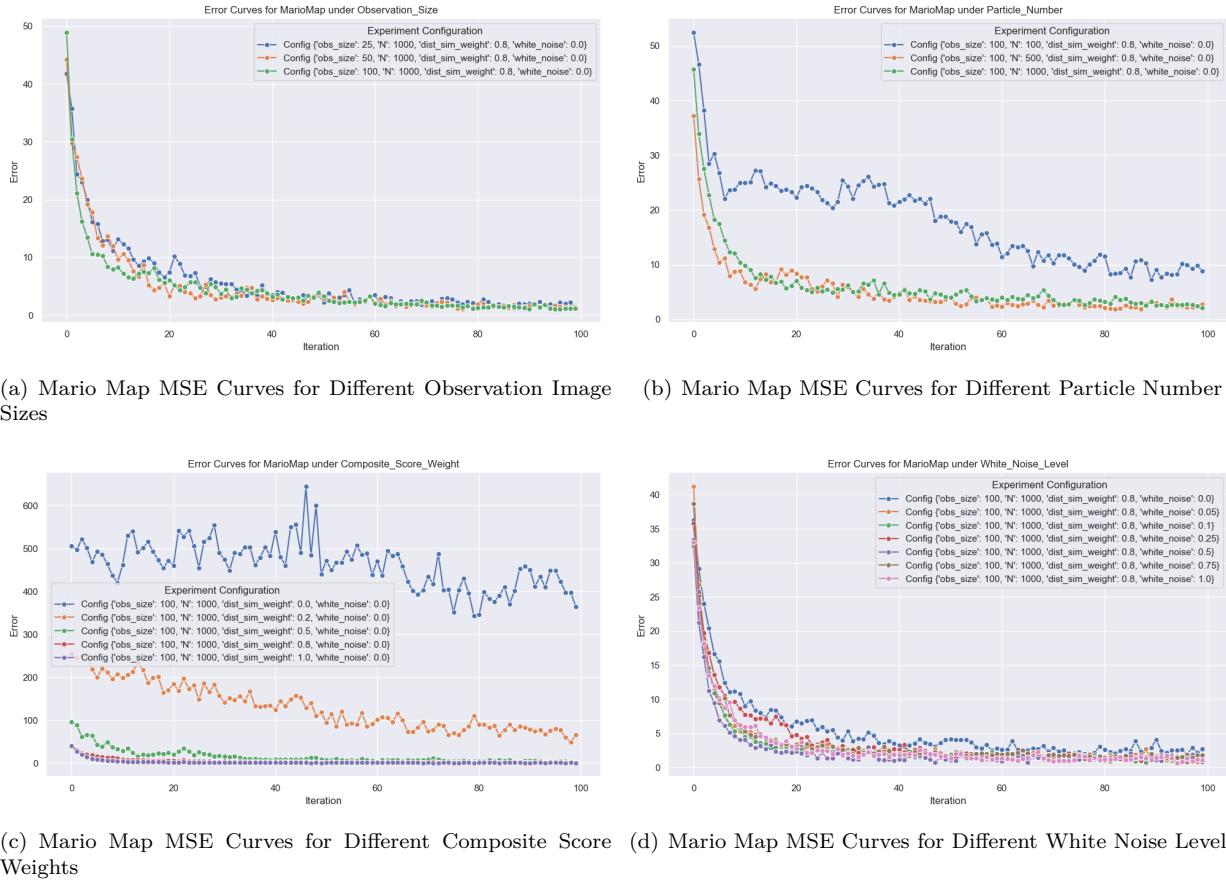


Figure 3: MSE Curves for Mario Map

As we can see from the results from running the experiments on the Bay map, the variable that influenced the error curve the most is the weight  $\gamma$  involved in calculating the composite similarity score between the

particles and the GPS location. As can be seen in 1(c), when  $\gamma$  is 0.0, the particles never converge to the true location. This is because when  $\gamma$  is 0.0, the similarity score becomes equal to the Structural Similarity Index (SSIM) of the particle's image and the reference image. In other words, when  $\gamma$  is 0.0, we don't consider the particle's closeness to the GPS location when assigning similarity scores anymore. This makes it hard to filter out particles far away from the true location as SSIM alone is not enough to distinguish a close particle from a far one. Many images far away from the true location can have high SSIM. For the Bay map, the particles tend to end up clustering on top of the ocean regardless of where the true location is when  $\gamma$  is 0.0 as shown in 3. The next variable that affects the error curve the most is the number of particles. When the number of particles is the lowest (=100), the errors are generally higher than when there are more particles. There are also more spikes in errors when the number of particles is small. This is not surprising as particles are initialized uniformly randomly and it is less likely that particles happen to be near the true location when there are fewer of them. Observation image sizes and white noise levels don't seem to affect the error curves too much. This adverse effect of reducing the number of particles is more pronounced on the Mario map as the Mario map is larger than the Bay map so particles are even more sparse. Generally, the variables' effects on the error curves are consistent across maps. As shown in 3(c), the SSIM measure seems to especially

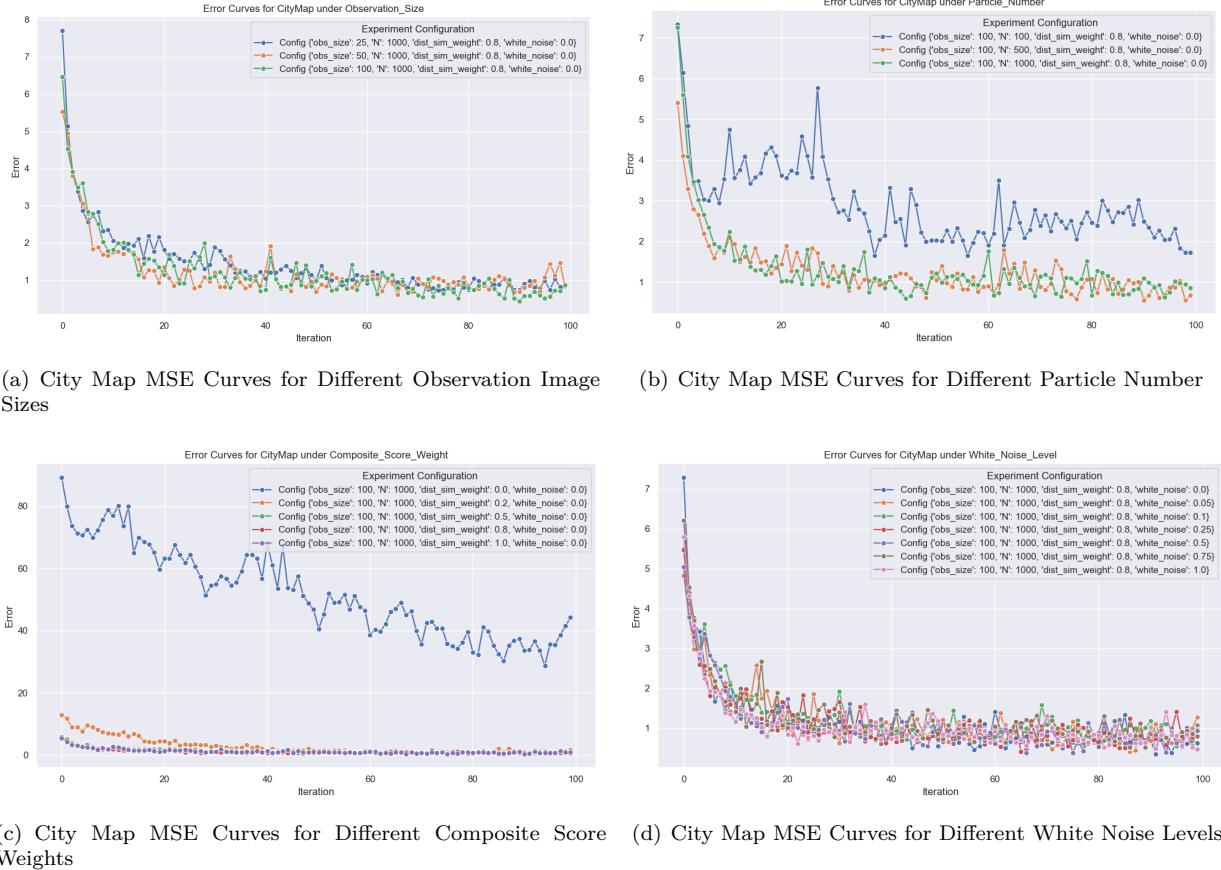


Figure 4: MSE Curves for City Map

struggle on the Mario map. Not only are the errors generally higher for lower  $\gamma$  values, but the particles also fail to cluster around the true locations for  $\gamma = 0.0$  and  $\gamma = 0.2$ . SSIM is based on a comparison of three key features: luminance, contrast, and structure. The Mario map is challenging because many observations have similar levels of luminance and contrast, therefore making it hard to distinguish which observation is closer to the reference image. In fact, the ocean in Mario map looks exactly the same everywhere whereas the ocean in Bay map has variations in both luminance and contrast.

White noise not affecting the error curves is not surprising. As the experiment with the composite score weight  $\gamma$  demonstrates, the distance-based similarity is the most helpful measure for distinguishing close particles from faraway ones. Since the composite measure is mainly relying on distance to assign particle

weights, adding white noise to the reference image should not affect the error curves.

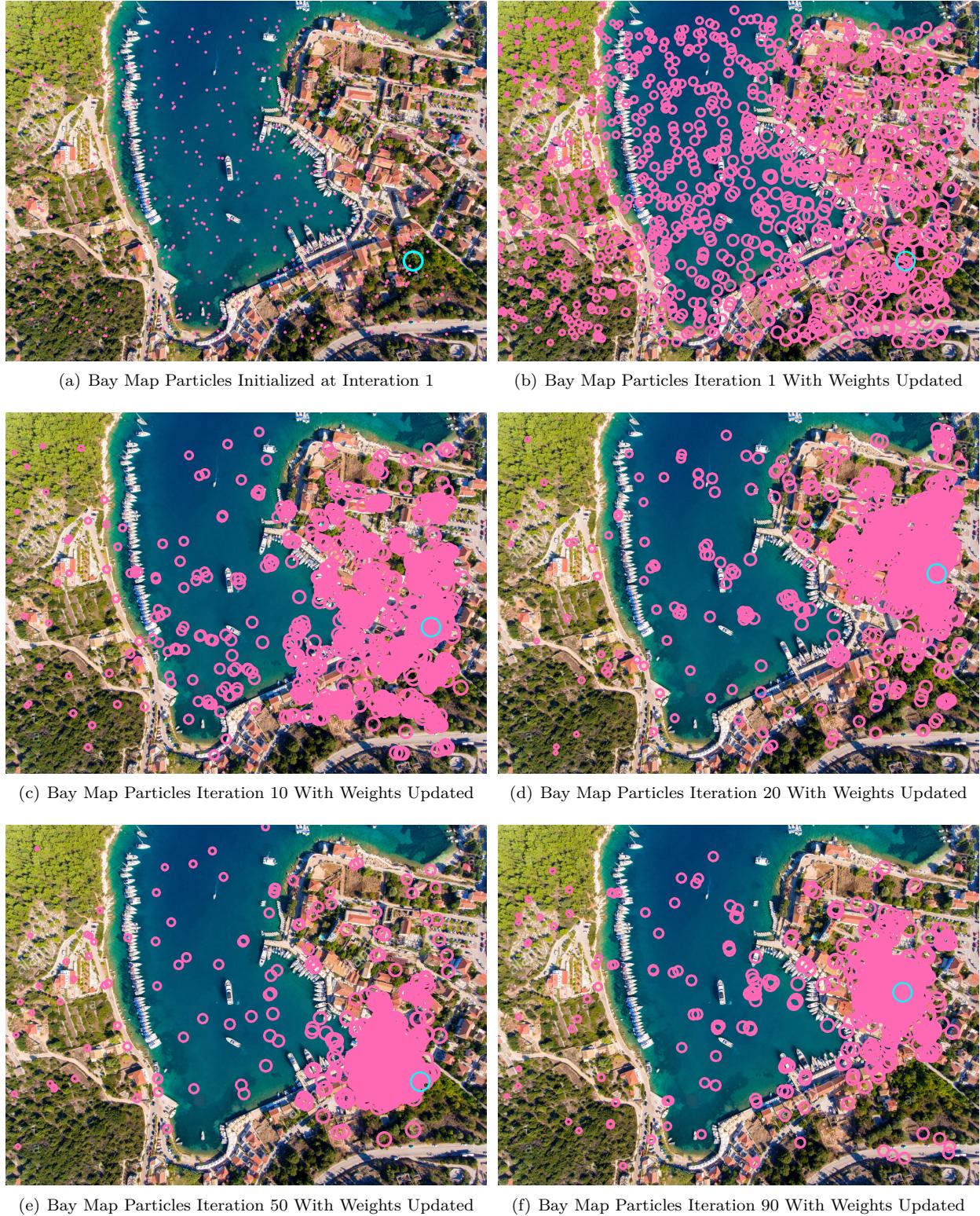


Figure 5: Bay Map Particle Filtering



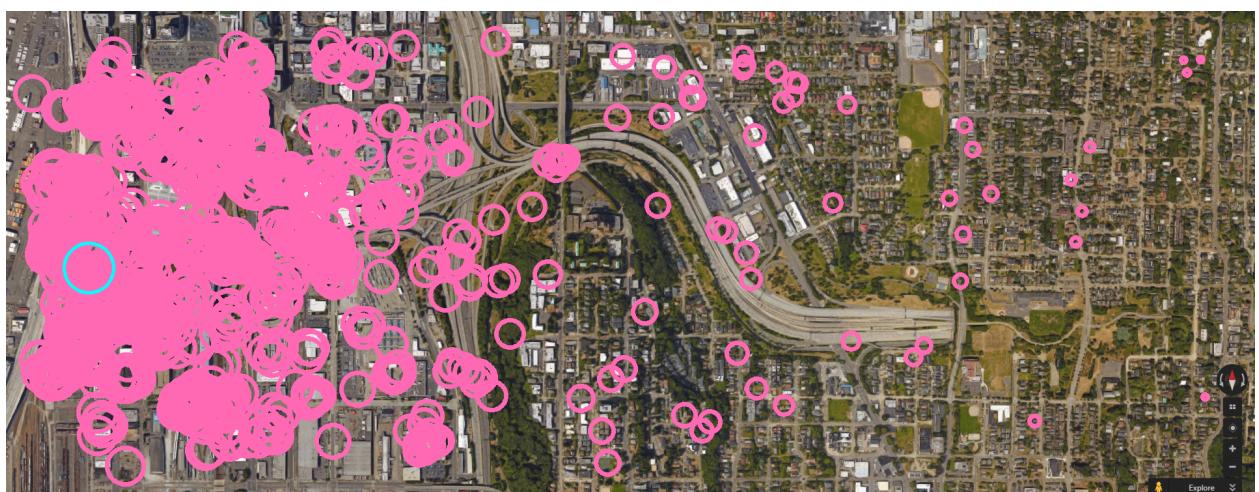
Figure 6: Mario Map Particle Filtering



(a) City Map Particles Initialized at Iteration 1

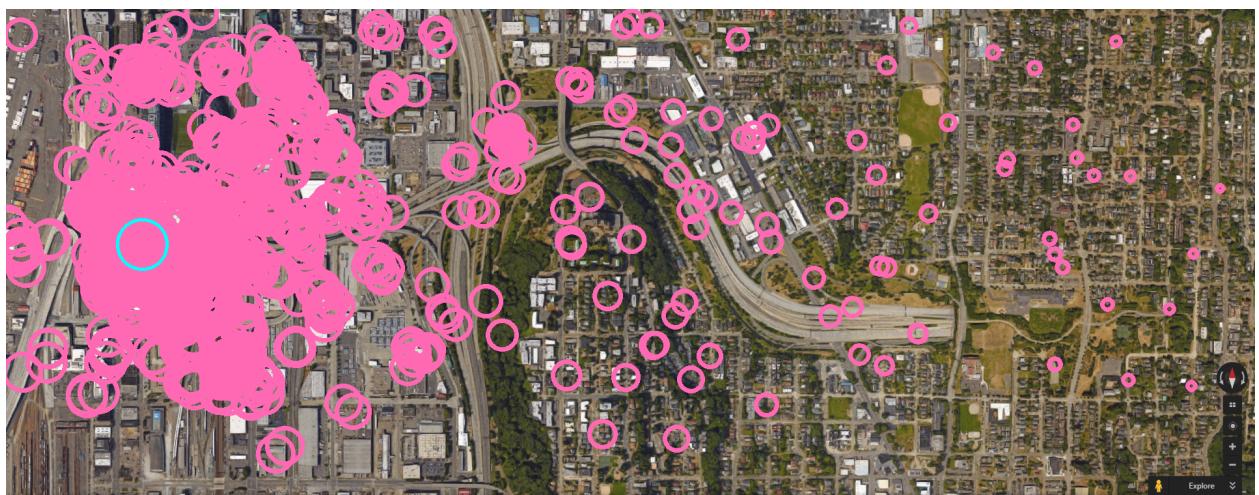


(b) City Map Particles Iteration 1 With Weights Updated

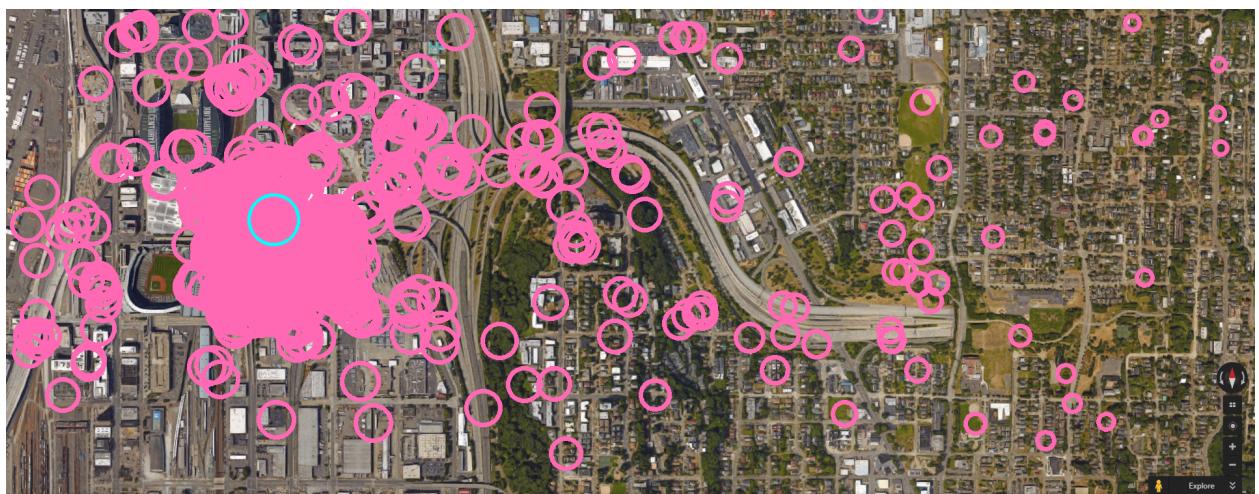


(c) City Map Particles Iteration 10 With Weights Updated

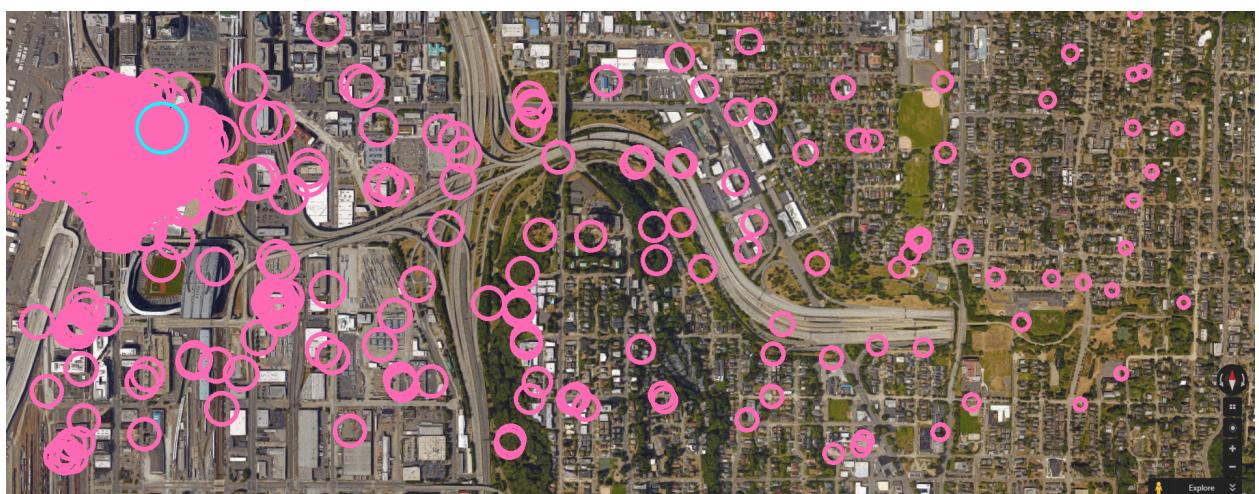
Figure 7: City Map Particle Filtering Iteration 1 - 10



(a) City Map Particles Iteration 20 With Weights Updated



(b) City Map Particles Iteration 50 With Weights Updated



(c) City Map Particles Iteration 90 With Weights Updated

Figure 8: City Map Particle Filtering Iteration 20 - 90