

Project Case Study

Luís Pedro Coelho

Programming for Scientists

March 17, 2009



University of Pittsburgh

Carnegie Mellon

First Steps for the Project

- 1 Email me your preferences
- 2 Get a google account
- 3 Send me the email for the google account (just email)
- 4 Wait for my confirmation
- 5 Checkout the repository (see google page)
- 6 Add yourself to AUTHORS.txt
- 7 Commit

Project Structure

- COPYING.MIT
- AUTHORS.txt
- README
- template.py
- setup.py
- docs/
- tests/
- particles/

```
-----  
This is a title  
-----
```

```
Now here comes my text. *This is bold*
```

```
Here's a list
```

- * first element
- * second element

```
def brownian_motion(mv_s,p_dis,crowding,crowding_std):  
    '''  
    particles = brownian_motion(mv_s,p_dis,crowding,crowding_std)
```

Generate particle tracks according to Brownian motion.

Parameters

- * `mov_sigma`: std. dev. for movement
(i.e., the position of a particle at time T
is given by a Gaussian centred on its position at time 0
with std. dev. `mov_sigma`)
- * `p_disappear`: probability that a particle disappears
- * `crowding`: expected number of particles
- * `crowding_std`: std. dev. of number of particles

The initial set number of particles is sampled from a normal distribution of mean `crowding` and std. dev. `crowding_std`. New particles are randomly generated to offset the dying particles and maintain the property

A product is only finished when you have the **software**, **documentation**, **tests**, and a **web-page**.

Project Goals

1 Generate a video of particles moving

2 Track the particles in this movie

3 Compare

4 Visualise

Project Goals

- 1 Generate a video of particles moving
 - a Generate tracks
 - b Generate video from tracks
- 2 Track the particles in this movie
 - a Detect particles
 - b Track particles
- 3 Compare
- 4 Visualise

Project Goals

- 1 Generate a video of particles moving
 - a Generate tracks
 - i Brownian motion (with parameters)
 - ii Brownian motion w. momentum
 - iii ...
 - b Generate video from tracks
 - i Shot noise (parameters)
 - ii Blur
 - iii ...
- 2 Track the particles in this movie
 - a Detect particles
 - i Global threshold
 - ii Local threshold
 - iii ...
 - b Track particles
- 3 Compare
- 4 Visualise

Project Goals

- 1 Generate a video of particles moving
 - a Generate tracks
 - i **Brownian motion** (with parameters)
 - ii Brownian motion w. momentum
 - iii ...
 - b Generate video from tracks
 - i **Shot noise** (parameters)
 - ii Blur
 - iii ...
- 2 Track the particles in this movie
 - a Detect particles
 - i **Global threshold**
 - ii Local threshold
 - iii ...
 - b Track particles
- 3 Compare
- 4 Visualise

Initial Guess

```
tracks = generate_tracks()  
video = generate_video(tracks)  
recovered = track(video)  
statistics = compare(tracks, recovered)  
print_statistics(statistics)
```