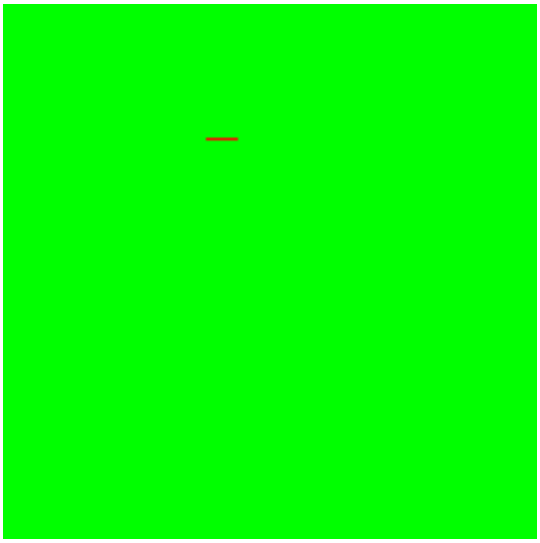
Particle filter - Track the snake

The goal of this practical study is to implement a simple particle filter with the programming language of your choice (Matlab or Python preferably). Two different models will be considered in part 1 and 2 respectively.

System description

This subject takes its inspiration from a simple video game available on some early mobile phones. A snake, 12 pixels long and with a thickness of one pixel, is moving in a plan 200×200 pixels wide. The snake is red and the background is green. An images sequence (numbered according to time t) showing snake moves is recorded in the folder *snake.zip*. You have to develop a program that detect and track the snake in the image during the whole sequence.



1 Simple model

As a first iteration of the developed algorithm, the estimated state (that is to say state vector represented by each particle) is the snake pixel position (x, y) (x et y are integer values). The simulated measurement to do is simple: if the current pixel of the estimated state is red (a part of the snake is here), the function return 1, and 0 else.The program has no prior knowledge about the snake initial position: A random sampling of the particles has to be done for the whole plane.

- Give the possible moves of the snake from one step t to the next one. Deduce a state transition model to code as a function.
- Implement a function that do the measurement for each state.
- Implement the general structure of the particle filter in order to estimate the snake position at each time t: define a set of particles, the initial sampling method, prediction step with the help of your transition state function, correction update with the help of the measurement function and the re-sampling process of your choice.
- particles positions has to be displayed as blue pixels in the images.
- Evaluation your solution by starting the algorithm at different times t of the sequence.
- Are we able to estimate precisely the snake's head thanks to this model?

2 More advanced model

We would like to ameliorate our model by estimating more precisely the snake's head. We define a model which take into account the head position and its orientation as a state vector. We modify the

measurement function so that it returns 2 if positions and orientations are verified, 1 only if position is well estimated, and 0 else.

Apply the developments and questions from the previous part to adapt your tracking function to this new model.