Representational Random Walks

Max Grusky
Oberlin College
mgrusky@cs.oberlin.edu

Abstract

By shortening and lengthening its individual segments in accordance with an underlying target image, we can encourage a random walk to reproduce the target image.

Introduction

Regular two-dimensional random walks produce interesting images by themselves. When bounded by the dimensions of a desired output image, they produce a complex marbled texture: darker in areas where the path has doubled back on itself frequently, and lighter in areas less traveled.

The goal of representational random walks is to manipulate the positioning of these lighter and darker patches so that they correspond with the light and dark portions of an image.

Method

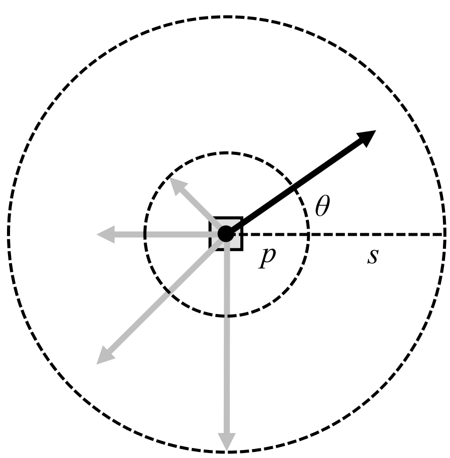

Figure 1 : *Varying segment length between* p *and* $(p+s)$ *based on the value of the current pixel. Other possible segments shown in gray.*

Representational random walks are composed of line segments connected to each other by random angles. If the random walk lands on pixel (i, j), it observes the brightness of this pixel. The pixel can range between $d(i, j) = 1$ if the pixel is perfectly dark, to $d(i, j) = 0$ if it is perfectly white.

Segment length depends on the step (s) and perturbation (p) values. Each segment's length varies linearly between p and $(p + s)$ based on the darkness of the pixel its starting point is currently sitting under, for $p > 0$. For $s = 0$, the representational random walk turns into a regular random walk. For p near zero, the walk spends more time filling in dark areas and jumping over light areas. Let (x_t, y_t) be the position of the random walk at time t. For a randomly generated angle $\theta \in [0, 2\pi)$:

$$x_{t+1} = x_t + ((1 - d(x_t, y_t))s + p) \cos\theta$$
$$y_{t+1} = y_t + ((1 - d(x_t, y_t))s + p) \sin\theta$$

A line segment is then drawn between (x_t, y_t) and (x_{t+1}, y_{t+1}), where (x_{t+1}, y_{t+1}) is bounded by the dimensions of the image.

The reason for introducing a perturbation value in addition to a step value is to handle the unfortunate situation where $d(x_t, y_t) = 1$, solid black. In this case, $x_{t+1} = x_t$ and $y_{t+1} = y_t$, and the random walk will get stuck in the "black hole." With zero perturbation, even patches of near-black can cause the path to slow down and take much longer to fill out the image. Adding a nonzero perturbation solves this problem.

Example: Random Dog Walk

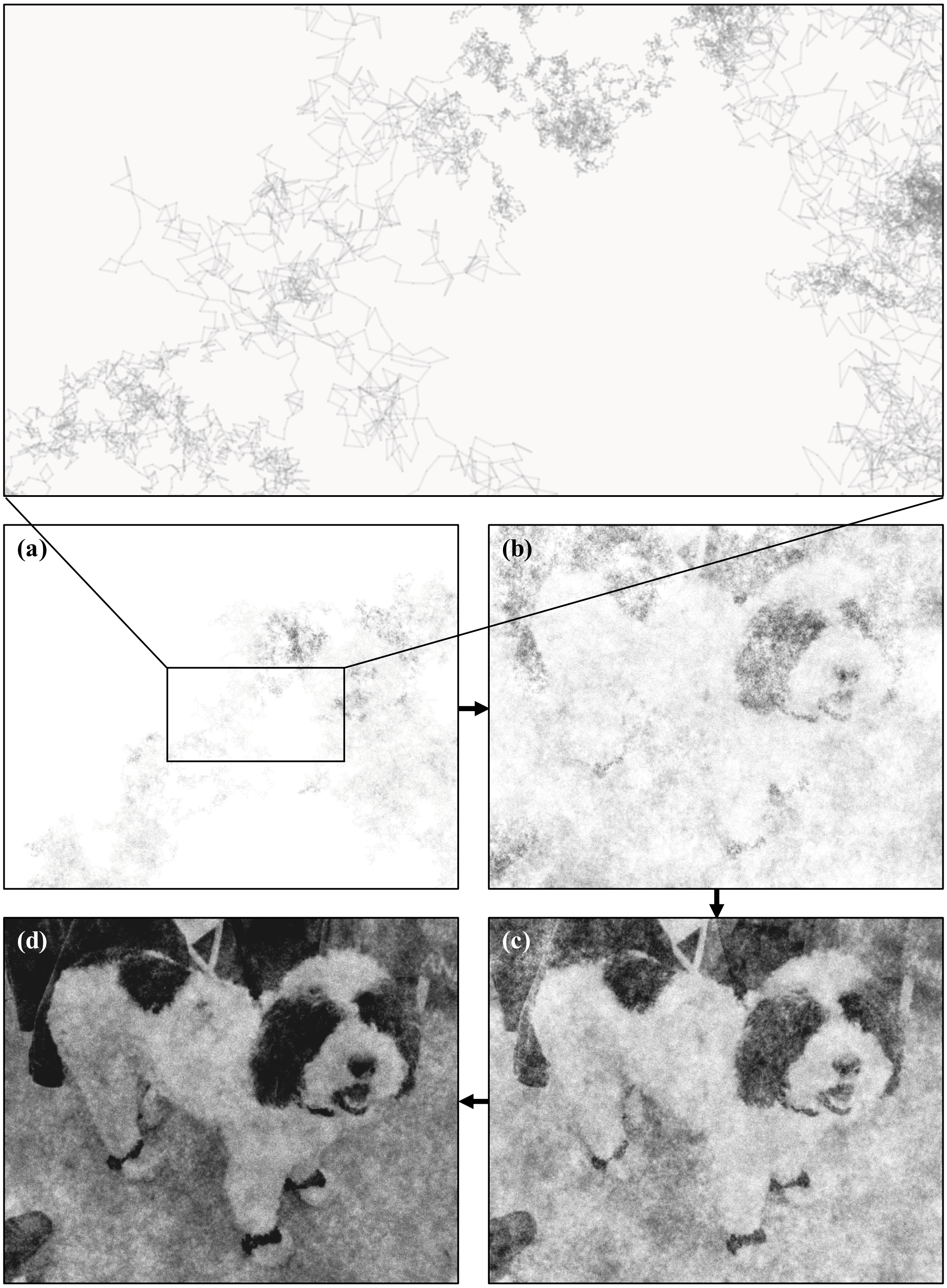

Figure 2: *Clockwise, starting from the top left. Representational random walks of (a) 100,000 segments, (b) 1,000,000 segments, (c) 5,000,000 segments, and (d) 10,000,000 segments. The expanded image at the top is darkened slightly to increase visibility.*

The examples above were produced using Processing [2]. The original image is 1000×800 pixels, with a step set to 10 pixels and a perturbation set to 1 pixel. A segment leaving a perfectly white pixel moves a distance of 11 pixels in a random direction, while a segment leaving a perfectly dark pixel moves a distance of 1 pixel. Finally, each line segment has a thickness of 1 pixel and an alpha value of 0.02, allowing for a larger variation in output values through overlapping lines.

Varying Step and Perturbation

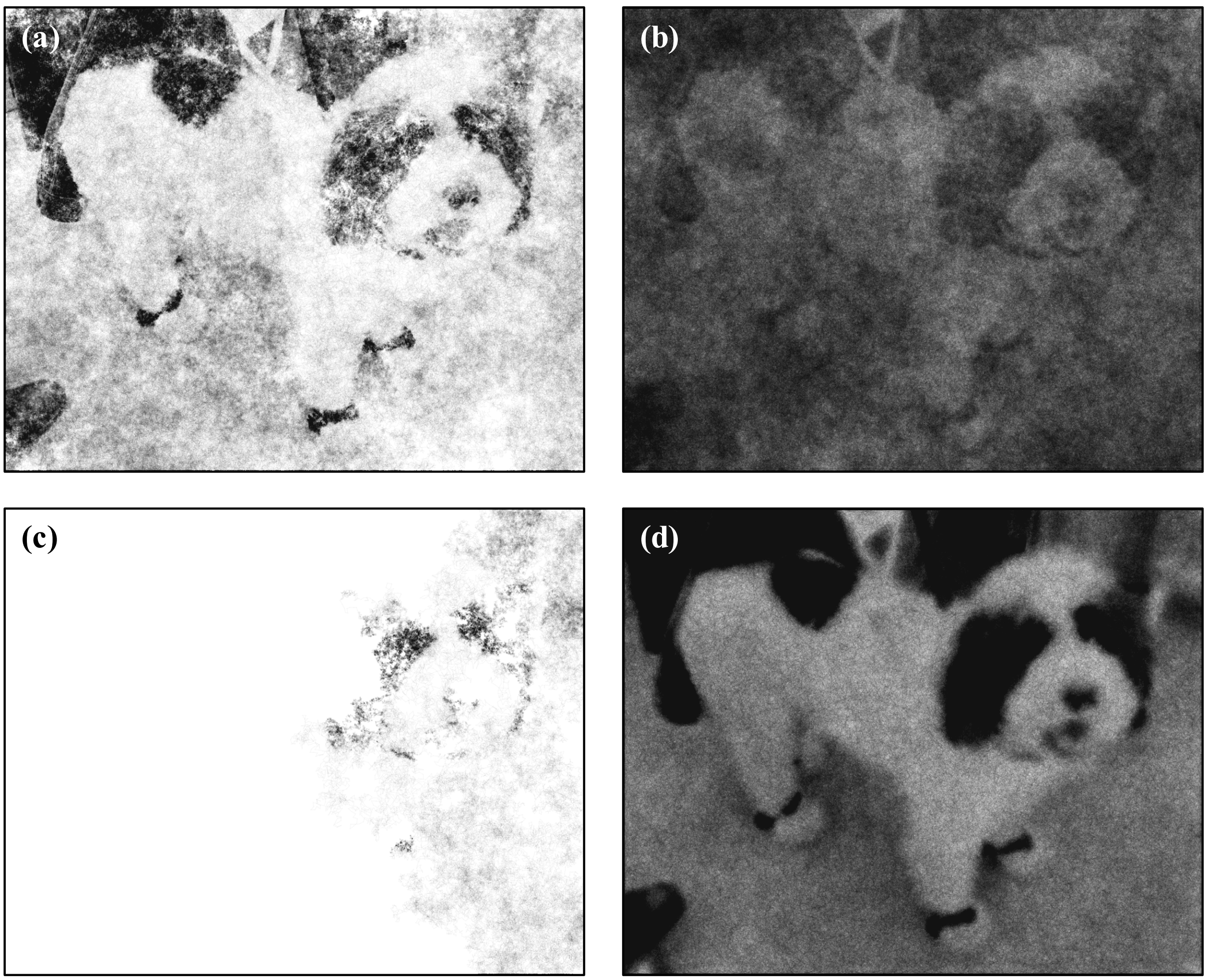

Figure 3: *Representational random walks with 5,000,000 segments where (a) $s = 10$, $p = 0.1$; (b) $s = 5$, $p = 5$; (c) $s = 10$, $p = 0.01$, and (d) $s = 50$, $p = 1$.*

Instead of varying the number of segments (t), these images show different combinations of step (s) and perturbation (p). Increasing perturbation increases the coverage of the image by allowing more random jumping regardless of pixel color, while decreasing perturbation increases image contrast by encouraging the random walk to stay within dark regions longer.

Increasing step size also increases image coverage, but decreases sharpness as individual line segments span longer distances and are less representative of the pixels directly underneath them. Decreasing step decreases the result's representativeness of the target image by making it act more like a typical bounded random walk. The amount of randomness of a representational random walk is best described by $s/(s+p)$, the fraction of a line segment's length that is determined by its starting pixel.

These four images were also drawn with a thickness of 1 pixel and an alpha value of 0.02. For source code and more examples using Processing, see [1].

Conclusion

By manipulating three parameters — step (s), perturbation (p), and number of segments (t) — a representational random walk can generate strikingly different reproductions of a target image. The images produced can have varying amounts of contrast, sharpness, and white space based on the configuration of the representational random walk. However, while representational random walk images are created through a random process, they are still visually similar to the original.

References

[1] Max Grusky. Representational random walks. `https://maxgrusky.com/posts/random-walks`, 2016. [As of March 15, 2016].

[2] Casey Reas and Ben Fry. Processing: programming for the media arts. *AI & Society*, 20(4):526–538, September 2006.