Informational Texture Synthesis

Sylvain Lefebvre

Informational Texture Synthesis

Sylvain Lefebvre
Inria

February 8, 2018

Abstract

This paper presents a novel technique to embed message into textures synthesized from tiles with boundary constraints. The messages are encoded in the choices that the synthesis algorithm normally makes at random during the synthesis process. Images embedding messages are not distinguishable from images that are not embedding any message. The random sequence seed used during synthesis can act as a password. This can be used similarly to a QR-code.

1 Introduction

This document is a work in progress and to keep it short and to the point I will skip introductory matters, bibliography and focus instead of the main principle.

The synthesis algorithm is based on the *wave function collapse* algorithm `https://github.com/mxgmn/WaveFunctionCollapse` which is itself a variant of prior algorithms (see references in the web page).

In the remainder it is to be understood that 'random' means 'pseudo-random using a sequential, deterministic generator'. Therefore, any random sequence produced during the algorithm can be exactly reproduced using the same start seed.

2 Synthesis

The core principle is to layout square tiles to form a large texture. Each tile can be seen as a label (the tile id). Tiles cannot be freely placed: constraints define how tiles can be neighboring (e.g. tiles A and B may only be placed with A to the left of B). This lets artists paint pattern pieces onto the tiles which, once assembled while enforcing all constraints form a consistent pattern e.g. a network of tubes.

Our variant of the algorithm goes as follows:

1. Initialize an array S of size $W \times H$, where each cell contains a vector of T booleans, with T the number of tiles. All vectors are initialized to *true*.

Each boolean represents the possibility that a particular tile appears at a particular location. Initially everything is possible.

2. Visit each location of S in some *fixed* order. For instance, one can use scanline ordering (left, right then top, bottom).

3. At each location, consider the vector of possibilities. If a choice exists (e.g. more than one tile is possible), we have an opportunity to embed a bit of information. A naive approach would select the first choice to represent '0' and any other choice to represent '1'. Once the choice is made, set all other entries in the vector to false.

4. Propagate the constraints to kill all choices in neighboring cells. This step is unchanged compared to the original algorithm.

5. Repeat until all cells of S have been visited. After this, a choice has been made everywhere and we can produce the final image by assembling the tiles.

This process can encounter a contradiction at step 3: in such cases no possibilities exist in the visited cell, due the propagation of constraints from previously visited tiles. If this happens, the algorithm is launched again with a different random sequence.

3 Decoding

The decoding algorithm starts by reconstructing the labels from the input image. This can be done by simple image comparison if the image is unaltered, or with some more elaborate vision based algorithm if the input comes from a photograph.

The decoder runs synthesis again. However, when a choice is possible, it uses the input to determine which choice was made during the initial synthesis. This decodes one bit of information each time (still assuming the same naive approach). At the end the bit stream is fully recovered.

If at some point the decoder finds a discrepancy (a set of choices that does not contain the tile selected in the input), then it can determine that synthesis failed and was re-run. It generates the proper state for the random generator and restarts.

4 Encoding information in choices

The initial proposal of encoding '0' in the first choice and '1' in any other is not a good strategy. Indeed, an empty message is represented by a zero bit stream: selecting consistently the first choice will strongly bias the synthesis process and could quickly run in contradictions. Instead, we map '0' to half the choices and '1' to the other half. Which half corresponds to '0' or '1' in randomly determined

at each cell. This approach is still limited to 1 bit per choice. Of course, when multiple choices are available the encoder can exploit this as additional storage space, in the manner of a data compressor (e.g. arithmetic coding).

5 Implementation

A proof of concept is available at `http://members.loria.fr/Sylvain.Lefebvre/infotexsyn/`. Code will be released when ready, but it should be simple to implement after reading this document.

6 Discussion

Tilesets This algorithm works with any tileset that provides choices (e.g. a tileset with a single tile provides none). Some tilesets will allow to encode larger messages than others in a same space. Whether and why a tileset allows for more information space in an intriguing theoretical question that we intend to explore. This can be measured experimentally by counting the number of choices offered by the tileset over a large number of synthesis.

Messages Another intriguing question is whether pathological messages exist, that would consistently lead to a contradiction regardless of the initial random seed. Such message would be impossible to encode.

Image size As it is difficult to ensure a message will fit, and since the synthesis is fast, we devise the following approach. First, the algorithm determines the size in bits of the message and uses previously computed average capabilities of the tileset to estimate the size of S (assuming some aspect ratio, e.g. 1:1). If the process terminates before all bits are encoded, S is enlarged and the process resumes. If on the contrary a lot of space is left empty, S could be reduced for a tighter fit. Note that messages are always padded with zeros until synthesis completes.

Steganography While this is clearly a form of steganography, it is unclear to this author whether it is a good or bad steganography approach, by lack of familiarity with the domain. The extent of the novelty is also an unknown quantity at this time.