Automated Key Generation in Texture Coding

Philip Ingrey

Texture coding [1] is the process by which an image, the 'plain-image', is encoded into a cipher-image for insecure transmission, c.f. Fig 1. This is carried out by taking the two-dimensional convolution of the plain-image with the so-called 'noise-field', a specially designed pseudo-random image used as the encryption key. The cipher-image obtained can then be distributed across an unsecured channel before being correlated with the noise-field to restore the plain-image.

Given that the cipher-image is made from a convolution of the original image with the noise-field, the information from the plain-image is distributed across the entire cipher-image. As such the cipher image can be degraded in many different ways, even including the removal of a portion of the image, and yet a visually similar image is still recovered after decryption.

Texture coding offers a robust system that could be widely employed, however the functions used within the algorithms, that generate the noise-field, are currently individually engineered. This leads to costly implementation and requires specialist expertise to run the system. It is this problem that the project seeks to address.

One method for generating the noise-field is to utilise a random number generator (the seed of the number generator forms the first part of the key) and to input the numbers generated into a known but highly-complex function (which forms the second part of the key) to generate the individual pixels of the noise-field. The function involved needs to be chosen such that its probability density function (pdf) is uniform over a range, otherwise a bias towards generating some numbers could be detected in the cipher-image and used to try and break the cipher. This project aims to expand the classes, and automate the generation, of functions that can be used for encryption purposes. The project will also seek to devise algorithms that are less onerous to implement and more robust to attacks.

The generation of highly-complex, or chaotic, functions with flat pdfs is a mathematically difficult inverse problem. As such this project will, in the first case, utilise a computational technique known as genetic algorithms to automate the generation of appropriate functions.

Genetic algorithms [2-3] are an iterative approach, useful in nonlinear optimization problems. During the first iteration a number of random functions are generated. With each successive iteration

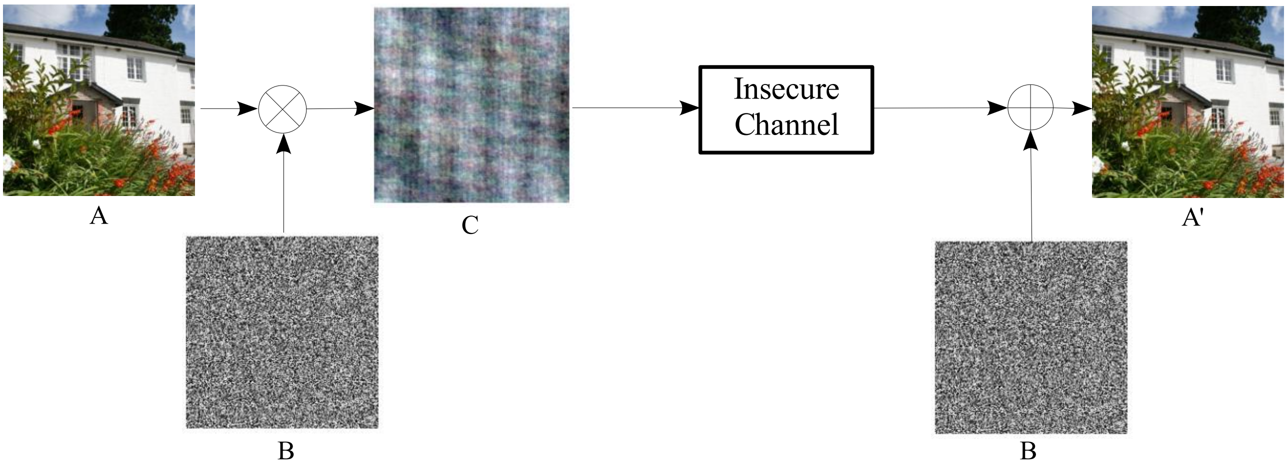

Figure 1: An illustration of Texture Coding. Following from the left; the plain-image, A, is convolved with the noise-field, B, to produce the encrypted cypher-image, C. At a later time the cypher-image can be correlated to the original noise-field, B, to reproduce the original image, A'. Here the same noise-field has been applied to each colour channel of A, however this need not always be the case.

each function is analysed to judge its fitness against criteria (here the function's complexity and the uniformity of the pdf). Subsequent functions are then generated based on the finest of the last generation. As the generation of functions progress strong candidates start to appear and are refined as further generations pass.

This project will seek to set-up the genetic algorithm code, benchmarking it first on a more straight-forward, but non-trivial, related problem of curve-fitting where the one seeks to minimizes the distance between a function and some given data points whilst also attempting to minimise the complexity of the function involved. This problem shares many points of commonality with, and will help inform, the full implementation. The complete texture coding algorithm will then be constructed, necessarily involving the production of methods for the rapid determination of a function's pdf.

Alongside the genetic algorithm approach other avenues of noise-field generation that are not predicated on chaos theory will be examined. One such system is a stochastic method for random process generation that produces processes with prescribed pdfs and first order correlation functions using the method described by Tough *et al* [4]. These processes can be used in place of the chaotic functions of the current approach. In so doing this will extend the class of functions that can be used in texture coding, and thereby increase the resilience to attacks.

Finally a complete evaluation of the robustness of each method will be considered, examining the statistics of the generated noise-fields, testing for any higher-order correlations that may give information on the encryption function used. If any correlations are found they will be used to re-inform the methodology employed improving the strength of the texture coding method.

This work differs from the PhD that is currently reaching its conclusion. Whilst the project will build upon the programming knowledge and expertise in stochastic processes developed in the PhD, it is in a dissimilar area and has a very different context and application. The PhD focused on the propagation of light in left-handed media while this project is concerned with image analysis and nonlinear optimization. Neither research into this type of image analysis or the application of genetic algorithms or stochastic processes to this area are currently undertaken in the School of Mathematical Sciences in the University of Nottingham.

[1] J.M. Blackledge and M.L. Hallot, "Covert Encryption and Document Authentication Using Texture Coding", *i-manager, Journal of Software Engineering*, 3, 1, 45 - 65, (2008)

[2] J.H. Holland, "Outline for a Logical Theory of Adaptive Systems", *JACM*, 9, 3, 297-314, (1962)

[3] D.E. Goldberg, "Genetic algorithms in search, optimization, and machine learning", *Pearson Education*, New Jersey, 1989

[4] R.J.A Tough and K.D. Ward, "The correlation properties of gamma and other non-Gaussian processes generated by memoryless nonlinear transformation", *J. Phys. D: Appl. Phys.,* 32, 3075–3084, (1999)