

Synthesis and Modification of Remotely Sensed Imagery

Stephen Brooks

Dalhousie University, Halifax, Nova Scotia, B3H 1W5, Canada
Telephone: +1-902-494-2512
Fax: +1-902-492-1517
sbrooks@cs.dal.ca, www.cs.dal.ca/~sbrooks/

1. Introduction

The impact of remotely sensed imagery is growing and its utility is reaching an increasingly broadly-based audience. The study of the earth extensively makes use of satellite imagery and most geospatial software systems incorporate it as a fundamental component. For example, 3D GIS such as TerraFly (Rishe et al., 1999) and GeoZui3D (Ware et al., 2001), offer the ability overlay satellite imagery onto terrains. In addition, textured terrains can also be used by GIS specialists as tools for the presentation of spatial data to non-experts (MacEachren, 1994). Recently, satellite imagery has extended to popular applications such as Google Earth.

The aim of this work is to increase the utility and flexibility of satellite imagery. The system being presented is able to completely synthesize new satellite imagery using data from existing images or seamlessly alter an existing image according to specified constraints. The proposed applications of this technology include the visualization of alterations to terrains, for both past re-construction and future prediction. Our system may also permit the generation of hypothetical geography for illustrative purposes in GIS education. The technical contribution of the work is a non-parametric method for the controlled synthesis of satellite images which combines pixel-based texture synthesis and non-linear image compositing.

2. The Concept of Image Re-synthesis

We wish to construct new satellite imagery incrementally and do so according to constraints. First, let us consider an example of synthesizing a new image based on data found in an existing image. In figure 1, a real satellite image is shown to the left. We will refer to this as the *input_image*. From the pixel data contained in this image we will construct the new image, shown in figure 2, left. We will refer to this as the *output_image*. Note that this second image is a fictitious terrain that does not exist.

There are two issues that must be explained. The first regards exactly how the pattern of the landscape is defined in the output image. As we will now discuss, the output image's terrain pattern is defined with a set of constraints which take the form of a *mask*. The second regards how the synthesis process is conducted, given the *mask* constraints. We will address this second issue in section 3.

In order to be useful, we need to be able to specify that a given portion of the output image will be a particular type of terrain. For example, in the *output_image* shown in figure 2, we have two explicit terrain types, namely water and land. We have used only two types for illustrative simplicity. The question becomes how to specify where the water is synthesized and where the land is synthesized.

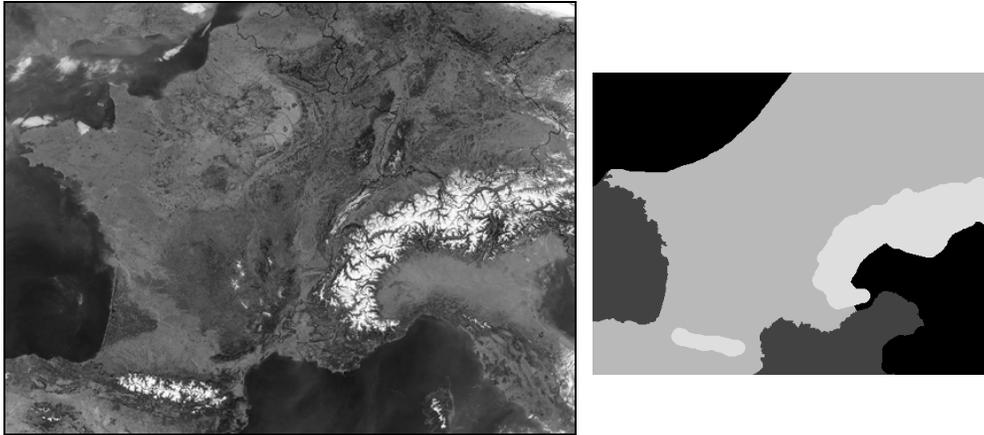


Figure 1: A real satellite image shown left. To the right is a mask that segments the satellite image into coherent regions.

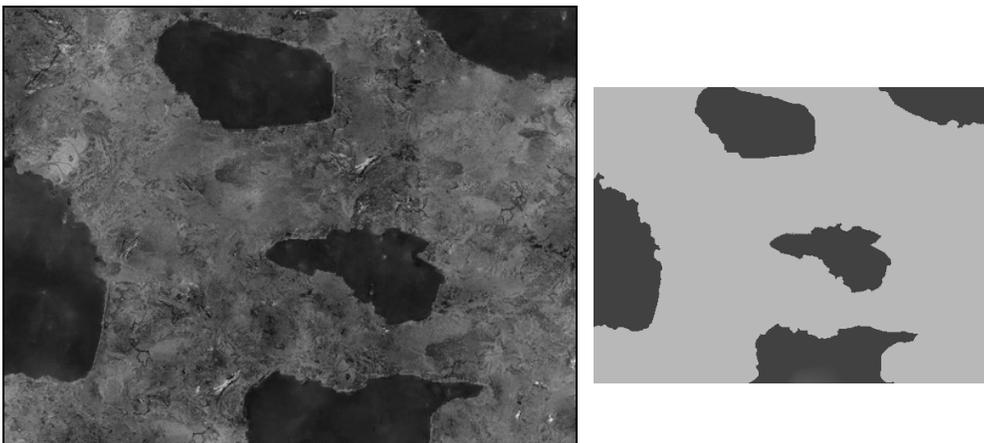


Figure 2: A newly synthesized image is shown left. To the right is a mask of the synthesized image that was used to constrain the synthesis process.

We can constrain the placement of terrain types with masks. Specifically, we require a mask for the input image (figure 1, right) and a second mask for the output image (figure 2, right). Note that the masks are shown smaller to fit the page width; in reality the masks and satellite images are the same dimensions. Informally, the input mask tells the system where each terrain type is “coming from”, while the output mask indicates where each terrain type is “going to”.

This set of four images form two analogous pairs. Specifically, we wish to find the analogous *output_image* that relates to the *output_mask* as the *input_image* relates to the *input_mask*. For now we will leave aside how the masks themselves are constructed, assume they are given and consider how the output image is synthesized under the mask constraints.

3. The Synthesis Process

We now propose a non-parametric method for the controlled synthesis of satellite imagery. This approach operates in two stages and combines pixel-based texture

synthesis and non-linear image compositing. By non-parametric, we mean that it does not require parameter tweaking to achieve high-quality results. This contrasts the earlier work on texture synthesis by Hertzmann et al. (2001) which requires a non-intuitive parameter to be set by the user, called the coherence parameter, κ . As our system does not require obscure parameters to be set by the user, it is easier to apply as it does not require the user to be familiar with the inner workings of the algorithm.

3.1 Generating an Initial Approximation

The first stage generates a rough approximation of the final result and could be considered a “base-coat”. Pixel-based texture synthesis is used to construct this approximation. We first produce a blurred version of the input image using a Gaussian lowpass filter of size 18 pixels with standard deviation of 3 pixels. Using the blurred input image, called *blur_image* (figure 3, left), we next generate the approximate solution, called *approx_image* (figure 3, right), using texture synthesis. We produce a blurred undercoat so that it does not introduce high-frequency artefacts into the final result.

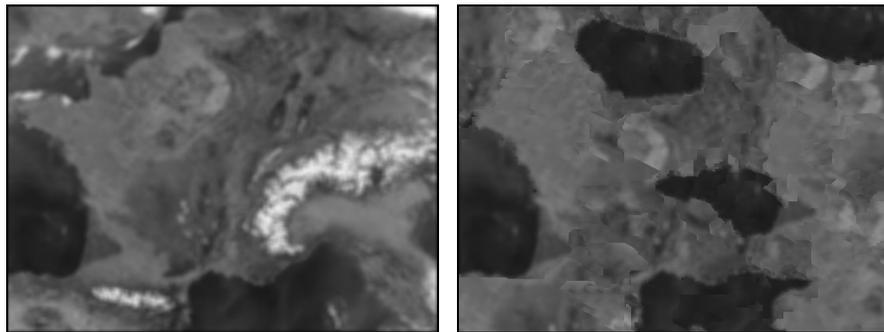


Figure 3: Blurred image satellite image (left). An initial approximation generated with pixel-based texture synthesis (right).

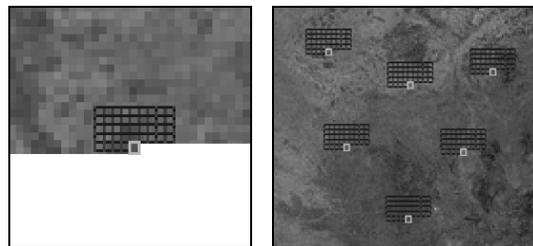


Figure 4: Synthesizing the next pixel (shown centre) in the output image (left). The next pixel is chosen from a selection of candidates with similar local neighbourhoods (right).

The algorithm generates each new pixel in the output image in sequential scan-line order. This is illustrated in figure 4 (left) with a newly synthesized pixel highlighted along with its local neighbourhood. To determine the new pixel’s colour, the pixel values of its local neighbourhood are compared against multiple local neighbourhoods in the input image (figure 4, right). The colour of the pixel with the best matching neighbourhood is used as the colour of the newly synthesized pixel. This pixel generation process is also subject to the constraints of the *input_mask* and

output_mask. For example, new pixels generated in a water region of the *approx_image* will only be sampled from a water area in the *blur_image*.

This discussion of the synthesis algorithm has been simplified for brevity. For more details the reader is referred to Hertzmann et al. (2001). In particular we use a fixed coherence parameter, $\kappa = 1$. Our results are not sensitive to the parameter since we are only using texture synthesis as a basis for the final image synthesis.

3.2 Constructing the Final Result

The *approx_image* contains the basic structure of the final image, but lacks detail and clarity. We now build directly upon the approximation. This phase requires an iterative algorithm which inserts detail into the *approx_image* using a sequence of non-linear optimization processes (Pérez et al., 2003).

But before this, we first need to identify region boundaries in the *output_mask*, using a Canny edge detector (Canny, 1986). The result of which is shown in figure 5, left. The edges are then expanded with a morphological dilation using a disk of diameter 10 pixels, shown in figure 5, right.

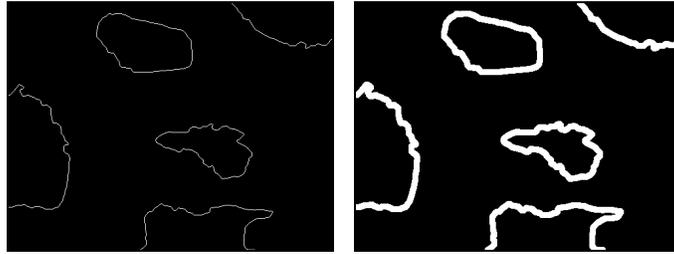


Figure 5: Canny edge detection on the *output_mask* (left) and subsequent dilation (right).

The next stage of the process is an iterative cycle which integrates detail into the non-boundary areas (shown black in figure 5, right) of the *approx_image*. At each iteration, details are extracted from a suitable region of the original *input_image* and seamlessly integrated into a subset of a non-boundary region of the *approx_image*. The algorithm iterates until all pixels within all non-boundary regions are processed.

Each iteration requires a separate optimization process to introduce new content into a subset, Ω , of existing the image, *approx_image* = h . It computes a new intermediate image, *inter_image* = f , whose gradient, ∇f , within Ω is closest to the gradient ∇g , taken from *input_image* = g . The original boundary, $\partial\Omega$, of region Ω from h is also used as a constraint. It is this additional constraint that ensures that the region Ω blends with the surrounding content, h . The resulting image contains an interpolation of the boundary conditions, $\partial\Omega$, inwards while conforming to the spatial changes of the guidance field from g as closely as possible within Ω . The minimization problem is written as:

$$\min_f \iint_{\Omega} |\nabla f - \nabla g|^2 \quad \text{with} \quad f|_{\partial\Omega} = h|_{\partial\Omega} \quad (1)$$

At the end of each iteration, *approx_image* is set to *inter_image*, meaning that each iteration builds detail onto the output of the previous iteration. After all iterations the final result is produced as shown in figure 2, left.

In this section we have generated an entirely new image. However, it is important to note that a more common application of this technology would likely be the modification of existing satellite images, in which case the *input_mask* and *output_mask* would be similar but not identical. We also note that the system can generate imagery with an arbitrary number of terrain types, rather than just the two (water and land) used in this proof-of-concept illustration.

4. Conclusions and Future Work

In this work we have achieved our first goal: to create a non-parametric system which generates new satellite imagery from exiting images, according to constraints. The proposed applications include the visualization of terrain change, for both past reconstruction and future prediction, as well as GIS education. But, there remains significant scope for extending the system.

At present the *input_mask* and *output_mask* are manually constructed. Further work is required to integrate image synthesis with complementary systems which would generate the constraint masks based on analysis and prediction. For example, a system which predicts land erosion could be adapted to automatically produce masks that would be used to generate the altered terrain image. More finely detailed control of the synthesis process could also be introduced. In practice, this would mean using continuously changing masks.

5. Acknowledgements

This work was supported by NSERC and the Canadian Foundation for Innovation.

6. References

- CANNY, J., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8, 6, pp. 679-698.
- HERTZMANN, A., JACOBS, C., OLIVER, N., CURLESS, B., and SALESIN, D., 2001. Image Analogies. Proceedings of ACM SIGGRAPH 2001, pp. 341-346.
- MACEACHREN, A. M., 1994. Visualization in Modern Cartography. ed. A. M. MacEachren and D. R. F. Taylor, Oxford, UK, Pergamon.
- PÉREZ, P., GANGNET, M., and BLAKE, A., 2003. Poisson image editing. ACM Transactions on Graphics, 22, 3, pp. 313-318.
- RISHE, N., SUN, Y., CHEKMASOV, M., ANDRIY, S., and GRAHAM S., 2004. System architecture for 3D terrafly online GIS. In Proceedings of Multimedia Software Engineering, pp. 273-276.
- WARE, C., PLUMLEE, M., ARSENAULT, R., MAYER, L. A., and SMITH, S., 2001. GeoZui3D: data fusion for interpreting oceanographic data. Proceedings of the MTS/IEEE Conference and Exhibition, 3, pp. 1960-1964.

Biography

Stephen Brooks is an Assistant Professor at Dalhousie University in Canada. Prior to this, he acquired a Ph.D. from the University of Cambridge in 2004 and a M.Sc. from the University of British Columbia in 2000.