An Approach to Synthetic Scene Completion and Periodic Noise Removal by Image Inpainting and Resynthesis

Alvin J. Spivey
Center for Imaging Science
Rochester Institute of Technology
11-19-2003

Credit Requirements: 2 Winter Quarter

Abstract

The Digital Imaging and Remote Sensing Image Generation (DIRSIG) model enables a user to generate synthetic scenes. With this package these scenes can span anywhere between .3—20 microns and encompass an array of spatial and spectral phenomena. One project facilitated by DIRSIG is megascene. In the megascene project projection of aerial imagery is used as a texture map. After scene rendering some areas are found to hold discontinuities along the ground due to these texture map projections. I propose to remove occlusions from rendered scenes. By applying an image inpainting algorithm missing regions are replaced with image surround through partial derivatives. Texture is then maintained with Harrison’s non-hierarchical procedure for re-synthesis of complex structures. The combination of these algorithms preserves scene continuity and accurate texture variability in the generation of any new synthetic scene. Additionally, a new approach to periodic noise filtering by image inpainting and re-synthesis is proposed.
1.) Introduction:

There are two applications of inpainting and texture re-synthesis as imagery solutions proposed here. Both of the applications capitalize on the biggest strength of the combination’s ability to replace any removal of an area with a close interpretation of the surrounding background. The combination’s solution to the megascene problem is to remove occlusions from a given texture map before its use for scene simulation, while maintaining surrounding background texture. The solution to periodic noise removal eliminates periodic noise through Fourier frequency domain inpainting and texture re-synthesis; preserving sinusoidal components of the given fast Fourier transformed image. In this proposal the origins of each problem is presented, along with their resolutions. The basics and theory of the image inpainting [2] algorithm and a non-hierarchical procedure for re-synthesis of complex textures [9] are covered. How an application of an inpainting and re-synthesis combination can be an answer to each problem is then surveyed, and proposed.

2. Background:

2.1 DIRSIG

Digital Image and Remote Sensing Image Generation model (DIRSIG) is an image modeling and simulation program package that enables a user to predict sensor reaching radiance field images between 0.3 - 20 microns [3]. These images are built from a variety of submodels (*i.e.* thermal, radiometric, etc.). DIRSIG, with these independent first principle physics based submodels, takes a ray tracing approach to making radiometric approximations. The user provides detector location and orientation, facet by facet spectrally attributed geometries and source position. As shown in Figure 1, rays
propagated from the source react with facets in the scene and then are propagated towards the sensor to produce a two-dimensional sensor-reaching radiance filed image [3].

2.2 Megascene

DIRSIG facilitates real world scene rendering projects such as megascene. The coverage employed by megascene spans a large area of several square miles, with significant spatial/spectral complexity. The scene consists of various types of houses, trees, grass, pools, asphalts, tennis courts, etc. that can be seen at .15 meter resolution. The spectral database currently has a spectral resolution of 1 nm and ranges from .4 to 2.5 microns [4]. Megascene is accountable for texture variability, shadows, thermodynamic property and leaf backscatter. However, one issue with the rendered images of megascene is unwanted occlusions. These occlusions in the texture map, some being rooftops and tree canopies, make the rendered scene spatially unappealing and may counter the radiometric fidelity of the area. The occlusions are initially wanted for use as a placement guide when setting objects (e.g. the projection of rooftops covered by house objects, the projection of tree crowns covered by tree objects, etc.). Once these objects are placed in the scene correctly, the texture map’s guides for object placements become occlusions; which need to be removed before scene rendering. One such rendered scene is shown in Figure 2, where rooftop and tree canopy occlusions remain. The tree canopy, along the right side of Figure 2, has a tree object planted over it, but still covers the sidewalk, grass patch, and reaches out into the road. The projected rooftop is missing a house object furthermore the projected rooftops with house objects on them can still be seen because the objects are not large enough. Before the project can be relieved of this problem four
goals must be achieved: remove occlusions, maintain scene continuity under conditions of great and low scene variability, and replicate texture for removed region.

Figure 1: The relative geometry between the sensor and the scene being imaged. The ray tracing approach of DIRSIG propagates rays from source, through the scene, and delivers radiometric information to the detector (http://dirsig.cis.rit.edu/about/overview/geometry.jpg) [3].

Figure 2: A rendered scene from the megascene project that consists of occlusions. The house rooftop is left uncovered in the center of the image and projected tree canopies lay across different areas of the scene (http://dirsig.cis.rit.edu/megascene/images/shot4-low.jpg) [4].
2.3 Periodic Noise Removal

The inpainting and texture re-synthesis combination is not limited to being purely an aesthetic tool for scene completion, but can also play as an analytical tool in Fourier space. This proposed imaging solution deals with the removal of periodic noise, like in Figure such and such, without the artifact of ringing, as seen with conventional periodic noise removal tools [8].

Traditionally the approach to ringing has been to convolve the impulse response due to noise with filters that have minimum sinusoidal components in the frequency domain [8]. Figure 3 shows three commonly used filters for edge detection and their spatial response after fast Fourier transform convolution in the frequency domain. While the Butterworth highpass filter and Gaussian highpass filter more closely approximate the impulse function in the frequency domain, they still exhibit some ringing [8].

With periodic noise the ringing subtends from the original noisy areas removed just like the edges in the convolutions of Figure 3. This could be prevented, not only by totally removing the noisy impulse function as is attempted with Fourier convolution of highpass filters [8], but maintaining continuity between the remaining sinusoidal components in the frequency domain while introducing no additional sinusoidal components into the frequency domain. When inverse transformed into the spatial domain the image will show no sign of periodic noise or ringing, and the original structure should be retained.
3.) Theory

The image inpainting algorithm was formed out of efforts to automate professional restorative techniques of paintings [6]. In the application of the algorithm the region is defined as in Figure 4. Figure 4 defines $\Omega$ as the region to be inpainted, and $\partial \Omega$ its boundary. After consulting conservators at the Minneapolis Institute of Arts the Bertalmio group found that there were four basic principles to retouching [1]:

---

Figure 3: Highpass filters and their spatial response. Because of the sinusoidal components necessary when convolving filters in the Fourier time domain ringing forms in images after inverse transformation. Each filter and their relative image convolutions are shown (IHPF—Ideal highpass filter; BHPF—Butterworth highpass filter; GHPF—Gaussian highpass filter). The first image is of the original.
1. The global picture determines how to fill in the gap, the purpose of inpainting being to restore the unity of the work
2. The structure of the area surrounding $\Omega$ is continued into the gap, contour lines are drawn via the prolongation of those arriving at $\partial\Omega$
3. The different regions inside $\Omega$, as defined by the contour lines, are filled with color, matching those of $\partial\Omega$
4. And the small details are painted (e.g. little white spots on an otherwise uniformly blue sky); in other words, “texture” is added.

The algorithm iteratively achieves goals (2.) and (3.) by smoothly propagating information from $\partial\Omega$ into $\Omega$ [7]. The algorithm however does not obtain the fourth principle—texture. It instead leaves the inpainted region structure coarse [6]. This problem is addressed with the addition of Harrison’s non-hierarchical procedure for re-synthesis of complex structures [9].

Figure 4: The Bertalmio et. al. definition of an inpainting region. Here $\Omega$ is defined as the region, and $\partial\Omega$ is defined as the region boundary.

Harrison’s approach to texture synthesis can be broken into 5 steps [6]:

1. Assign information content to each pixel based on neighboring pixels (this weighting determines which pixels will be chosen)
2. Find pixel with matching neighbors by Manhattan Distance “city block” measure [9]
3. While there exist undetermined pixels, choose the highest priority pixel based on weighting found in (1.)
4. While there exist undetermined pixels choose the closest distance measure of (2.)
5. Define digital count and update neighboring pixels’ weights.

3.1 Image Inpainting

To explain the image inpainting algorithm Bertalmio defines an image as follows [2].
$I_0(i, j) : [0, M] \times [0, N] \rightarrow \mathbb{R}$, with $[0, M] \times [0, N] \subset N \times N$, is a discrete 2D gray level image component that make up the initial image $I_0$, with pixel location $(i, j)$. The Bertalmío image has dimensions defined by columns $M$ and rows $N$ that are a set of real numbers. The color components (R, G, and B) are then represented as multiple $N \times M$ grayscale images separately of the same $M \times N$ dimension. A group of images $I(i, j, n) : [0, M] \times [0, N] \times \mathbb{R} \rightarrow \mathbb{R}$ are created with each step $n$ of the inpainting algorithm.

Here the input image is defined at $n=0, I(i, j, 0) = I_0(i, j)$, and it is shown that iteratively with each step $n$ an answer is achieved, $\lim_{n \to \infty} I(i, j, n) = I_R(i, j)$. The algorithm expressed in general form is then:

$$I^{n+1}(i, j) = I^n(i, j) + \Delta t I_R^n(i, j), \forall (i, j) \in \Omega.$$  \hspace{1cm} (A.)

This general algebraic expression for the inpainting algorithm is all happening within the missing region $\Omega$, the superscript $n$ is an integer step for updating the image $I^n(i, j)$, $\Delta t$ the unit less rate of improvement and each update to the original $I^n(i, j)$. With every $n$ the image $I^{n+1}(i, j)$ becomes closer to filling the missing region. This update, $I^n(i, j)$ is better defined.

To accomplish principles (2) and (3), lines along the side of the region $\partial \Omega$, must be continued through as isophotes (i.e. lines of same grayscale values) and scene information outside the region, $\Omega$ must be brought into the region. Let $L^n(i, j)$ be the scene information and $\overrightarrow{N^n}(i, j)$ the direction scene information reaches into $\Omega$. An expression related to the updated amount is then generated as,

$$I^n(i, j) = \delta L^n(i, j) \cdot \overrightarrow{N^n}(i, j).$$ \hspace{1cm} (B.)
In the definition of update, \( \delta L''(i, j) \) is the measure of change in information \( L''(i, j) \). In the implementation of equation B, \( \delta L''(i, j) \) is an estimation of scene information by the Laplacian filter, \( \nabla^2 \). Knowing this, equation B becomes more familiar as the diffusion equation classically used for heat transfer in differential equations [10]. Where by construction of Equation B, \( \delta L''(i, j) \) is replaced by the initial concentration of the diffusion equation \( U(r, t) \), and just as concentration at some distance \( r \) away is calculated, \( I''(i, j) \) at some point \( n \) is determined.

\[
U_i(r, t) = \alpha \nabla^2 U(r, t) \tag{C.}
\]

\[
\nabla^2 = \sqrt{\left( \frac{\partial^2 x^2}{\partial t^2} + \frac{\partial^2 y^2}{\partial t^2} \right)} \tag{D.}
\]

\[
\delta L''(i, j) := (L''(i + 1, j) - L''(i - 1, j), L''(i, j + 1) - L''(i, j - 1)) \tag{E.}
\]

\[
L''(i, j) = I''_{xx}(i, j) + I''_{yy}(i, j) \tag{F.}
\]

Figure 5: Laplacian kernel [7].

\[
\begin{array}{ccc}
0 & 1 & 0 \\
1 & -4 & 1 \\
0 & 1 & 0 \\
\end{array}
\]

The classical differential diffusion equation for heat transfer, where the concentration, \( U_i(r, t) \), at some distance \( r \) away is proportional, \( \alpha \), to the product of the Laplacian, \( \nabla^2 \) and original concentration, \( U(r, t) \), is shown below. The Laplacian defined by partial derivative addition, \( \sqrt{\left( \frac{\partial^2 x^2}{\partial t^2} + \frac{\partial^2 y^2}{\partial t^2} \right)} \), is in Equation D. The calculation of the
amount of scene information $\delta L^n(i,j)$ by the Laplacian $L^n(i,j)$ is shown through

Equations E-F, where $I^n_{xx}(i,j) + I^n_{yy}(i,j)$ is the sum of squared partial derivatives in the x- and y-direction, respectively. Figure 5 shows the corresponding Laplacian kernel.

$$
|\nabla I^n(i,j)| = \frac{\partial I}{\partial t}(x,y,t) = g_e(x,y)k(x,y,t)|\nabla I(x,y,t)|, \forall (x,y) \in \Omega^e
$$

(G.)

$$
\vec{N}(i,j,n) = \frac{(-I^n_x(i,j),I^n_y(i,j))}{\sqrt{(I^n_x(i,j))^2 + (I^n_y(i,j))^2}}
$$

(H.)

$$
I^n_t(i,j) = \left(\delta L^n(i,j) \cdot \frac{\vec{N}(i,j,n)}{|\vec{N}(i,j,n)|}\right) |\nabla I^n(i,j)|
$$

(I.)

Figure 6: Kernel for gradient $\left(I^n_x(i,j) + I^n_y(i,j)\right)$ of equation H.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>-</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

Consider that $\vec{N}(i,j)$ only holds directional information, not curvature. Equation B must be rewritten using the normalized magnitude of $\vec{N}(i,j)$ and the directional gradient vector $|\nabla I(x,y,t)|$ through use of the equation for anisotropic diffusion, Equation G. In the anisotropic diffusion equation the directional gradient vector, $|\nabla I(x,y,t)|$, is taken into account, and $\Omega^e$ is a dilation of the inpainting region, $\Omega$, with a ball of radius $\epsilon$, $k$ as the Euclidean curvature of the isophotes of the image. $g_e(x,y)$ is a smooth function in the shrinking inpainting region, $\Omega^e$, such that $g_e(x,y) = 0$ along the edge of the shrinking inpainting region, $\partial \Omega^e$, and $g_e(x,y) = 1$ in $\Omega$. This allows the normalized
magnitude of \( \vec{N}(i, j) \) to be determined, properly angled by \( |\nabla I(x, y, t)| \) through anisotropic diffusion, and propagated into \( \Omega \) from \( \partial \Omega \) correctly. Equation H defines the normalized magnitude of \( \vec{N}(i, j) \) as defined by the image gradient components \( I^n_y \) and \( I^n_x \), and Equation I is the revision of equation B that is now accountable for isophotes curvature. Figure 6 shows the gradient kernel used in Equation H.

\[
\beta^n(i, j) = \delta L^n(i, j) \cdot \frac{\vec{N}(i, j, n)}{|\vec{N}(i, j, n)|} \tag{J.}
\]

\[
|\nabla I^n(i, j)| = \begin{cases} 
\sqrt{(I^n_{xhm})^2 + (I^n_{xfm})^2 + (I^n_{yhm})^2 + (I^n_{yfm})^2}, & \text{when } \beta^n > 0 \\
\sqrt{(I^n_{xhm})^2 + (I^n_{xfm})^2 + (I^n_{yhm})^2 + (I^n_{yfm})^2}, & \text{when } \beta^n < 0 
\end{cases} \tag{K.}
\]

Depending on the amount of information in any given direction the inpainting algorithm is subject to change position. With \( \beta^n \) defined as the projection of \( \vec{\delta L} \) onto the (normalized) vector \( \vec{N} \), otherwise known as the change of \( L \) along the direction of \( \vec{N} \), the gradient vector decides whether to continue on the current path (\( \beta^n > 0 \)) or change direction (\( \beta^n < 0 \)), and if so by how much. Equations J and K show these propagation parameters.

3.2 A Non-hierarchal Procedure for Re-synthesis of Complex Texture

The re-synthesis procedure assumes pixel independence, and that each color component (R, G, and B) is a sum of normal distributions. Using a weighting scheme, the relationship of each pixel \( I(i, j) \) within \( \Omega \) is explored and used as an order of significance
and in the Manhattan distance equation [9]. Manhattan distance measure is used to
define the region to be prioritized. A table of symbols and the Manhattan distance
measure, Equation N, are shown below.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Omega_I )</td>
<td>The set of pixel locations in the input image.</td>
</tr>
<tr>
<td>( I(s), s \in \Omega_I )</td>
<td>The pixel in the input image at location ( s ).</td>
</tr>
<tr>
<td>( \Omega_O )</td>
<td>The set of locations containing pixels in the output image.</td>
</tr>
<tr>
<td>( O(s), s \in \Omega_O )</td>
<td>The pixel in the output image at location ( s ).</td>
</tr>
<tr>
<td>( L(s), s \in \Omega_O )</td>
<td>The location of the pixel in the input image that is the output image at location ( s ).</td>
</tr>
<tr>
<td>( \Omega_K )</td>
<td>The set of offsets considered when calculating the similarity of two texture patches.</td>
</tr>
<tr>
<td>( K(u), u \in \Omega_K )</td>
<td>Weighting given to a particular offset ( u ).</td>
</tr>
<tr>
<td>( A(s), s \in \Omega_I )</td>
<td>Two uniformly random functions.</td>
</tr>
<tr>
<td>( B(s), s \in \Omega_O )</td>
<td>Two uniformly random functions.</td>
</tr>
</tbody>
</table>

\[
D(s,t) = \varepsilon |A(s) - B(t)| + \sum_{u \in \Omega_K, t+u \in \Omega_O} K(u) d(I(s + u), O(t + u)) \tag{N.}
\]

Only pixels \( \Omega_K \) are considered for weighting. Each \( \Omega_K \) pixel \( I(s) \) and its neighbors at an
offset \( I(s+u) \) are considered as a normally distributed. The bits of information \( G(s,u) \) are
calculated by three parameters. Three parameters are found by: (i.) number of bits \( m \)
required to encode pixels of each family; (ii.) families then classified by the most \( m \)
significant bits; and (iii.) standard deviations for each pixel are found. Image entropy, or
the number of bits needed to store normally distributed variables with a given standard
deviation \( \sigma \), to a \( \pm \frac{1}{2} \) accuracy is computed as,

\[
H_{image} = \frac{1}{2} \log_2(2\pi) + \frac{1}{2} \ln 2 + \log_2 \sigma \tag{O.}
\]

\[
G(s, u) = H_{image} - H(s, u) \tag{P.}
\]
The number of bits given by $I(s)$ within its family can then be found as the difference between entropy of the family of $I(s)$ and the entropy of the pixels in the image. This weight $G(s, u)$ is shown in Equation P. $G(s, u)$ is then normalized and ranked as shown in Equation Q, and the Manhattan distance weight variable $K(u)$ from Equation N is defined in Equation R. In Equation R, $G(u)$ is the average value of the weightings $G(s, u)$.

4. Methods and Approach

The $\Omega$ is provided as input, a mask of the occlusion area noted by 1’s, with 0’s elsewhere. Intermittent runs of inpainting and anisotropic diffusion are made throughout the $\Omega^E$ substitution. However, this substitution does not achieve the fourth principle (4.): the small details are painted (e.g. little white spots on an otherwise uniformly blue sky); in other words, “texture” is added. To achieve the fourth principle Harrison’s re-synthesis procedure is applied. Figure 8 is an example of what image inpainting produces and how re-synthesis helps.

*Figure 8: Input image with mask of inpainting area, inpainted region product, and re-synthesized image (from left to right). Note how the inpainting delivers a coarse substitution of $\Omega$ [6].*
Combined, the algorithms achieve each of the four goals needed to solve the megascene occlusion problem, as well as cover the issues with periodic noise removal.

The combination will be applied as a stand-alone image tool. The GUI of the stand alone will accept and display the given original image, the mask, and eventually an inpainted and textured output. The user will have the choice of running the algorithm in memory or having the output saved to a file. All standard image formats will be acceptable.

4.1 Megascene

The combination will act as a texture map preprocessing tool for the megascene solution. The texture map occlusions will be removed after the objects are planted. The success of the application will be measured by observation. The occlusion removal success will be measured by how well the goals necessary for completion are accomplished: removal of occlusions, maintenance of scene continuity under conditions of great and low scene variability, and replication of texture for removed region.

4.2 Periodic Noise

Input for periodic noise removal will be the Fourier spectrum of the image and a corresponding mask of impulse functions. To test the ability of the combination to remove periodic noise without artifacts, high pass filtering of images in the time domain
defined by circles that have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively [8] will be tested for ringing when applying inpainting. A comparison of conventional filters to the algorithm will be made by root mean square error (RMSE), where at the edges of the image values would be smallest error. The best technique that has the best fit to the images edge would then have the smallest RMSE region about the edges. Figure 9 shows the image and its Fourier spectrum with the defined circles.

Figure 9: The test target and its Fourier spectrum. The circles have radii of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively [8].

5. Resources:

The algorithm and GUI will be coded in IDL 5.6, and is dependant on user input of original image and a digitally edited mask of undesired section. Both the input image and mask must have corresponding dimensions. The test images for periodic noise removal
will come from the digital image processing, 2/E image database (www.prenhall.com/gonzalezwoods).

6. **Timetable**

Research background of Image Inpainting

Design IDL program to compute and display Bertalmio algorithm

Develop a (GUI) to display and output results

Integrate algorithm into DIRSIG

Research poster preparation

Research presentation preparation

Senior thesis

7. **Budget:**

2 Winter Quarter Credit Hours at $10.00 an hour.

8. **References:**


