COMPRESSED POINT CLOUD TERRAIN MODELS USING WEDGELETS

CHRISTOPHER MILLER

Abstract. In this paper, we propose a technique for generating digital elevation models (DEM) of urban terrain from raw LIDAR point cloud data using an image processing technique, called wedgelets, developed by Donoho. The advantage of this technique is that the models it generates are very efficient in terms of storage demands, and the technique is highly resistant to noise. We begin by providing a mathematical justification for the use of wedgelets on urban data and validate our technique by using it to create DEMs using both an urban and non urban LIDAR point cloud.

1. Background

The use of aircraft-based LIDAR systems has become an important tool for creating elevation models of terrain data. To summarize the method: an aircraft is equipped with a laser range finder and a GPS system. The laser range finder calculates the ground elevation while the GPS system calculates the location of the aircraft over the surface of the earth. The combination of these two data streams produces an elevation model of a given area consisting of a large number of discrete points in three-dimensional space. This set of points is called a point cloud.

The problem of producing an image of the terrain from this point cloud is subtle and difficult. The first problem is that the point cloud is non uniformly sampled. This is due to altitude and airspeed changes, as well as changes in the local curvature of the terrain being sampled. This problem makes typical multiresolution analysis difficult due to the varying density of information throughout the nested subspaces.

The second problem is that due to various issues arising from the physical implementation of the LIDAR system, the resulting point clouds tend to contain a relatively high level of noise. If the reflectance or refraction index of a particular region being struck by the laser is an extreme value, incorrect measurements may be obtained.

Most methods for creating a representation of the terrain from the point cloud data focus on using interpolative processes to approximate the terrain locally, using low degree polynomials on arbitrary meshes. These techniques have are disadvantaged in that they do not admit a compact representation. Since the mesh is arbitrary, the sender needs to transmit the mesh information as well as the local polynomial coefficients. Other methods, such as LOESS, require the original data set to be present in order to reconstruct the visualization. All of these methods prevent the real time transmission of terrain representations obtained through LIDAR point cloud sampling.

The objective of this project is to develop software that will create a visual representation of urban terrain using a LIDAR point cloud. The method should be able to cope with the difficulties of non uniformly sampled data and noisy data. The most important aspect, however, is that the representation must be compact. This will enable real time transmission of the representation.

2. Wedgelets

2.1. **Justification for Development.** The distinguishing feature of urban terrain data is the presence of very strong edges. Therefore, we must implement a technique designed to handle discontinuous data. Wavelets have demonstrated excellent ability at decomposing piecewise smooth, one dimensional signals with discontinuities. In particular, the local support of the wavelet basis functions allows wavelets to efficiently represent discontinuities in the signal. Wedgelets are shown to be a class of approximating functions that perform optimally in the minimax sense on a class of data similar in principle to urban data.

Definition 1. *Let X be a set of signals (or images). An estimator $\delta^m : X \to \Theta$ is called minimax with respect to a risk function $R(\theta, \delta)$ if*

$$sup_{\theta\in\Theta} R(\theta, \delta^m) = inf_{\delta} sup_{\theta\in\Theta} R(\theta, \delta).$$

The risk function we will be considering will be the L_2 error between an image in the class X and its approximation θ.

Consider an orthonormal basis $(\psi_\lambda)_\lambda \in \Lambda$ for $\mathbb{V}$, the set of piecewise real (or complex) valued C^2 functions on $[0, 1]$. For any $f \in \mathbb{V}$ and $n \in \mathbb{N}$ let $\epsilon_n(f)$ be the best nonlinear approximation of f using a linear combination of N basis elements.

$$\epsilon_n(f) = inf\{||f - \sum_{\lambda\in\Lambda'} a_\lambda \psi_\lambda||^2 : |\Lambda'| = n\}$$

It can be shown that the approximation error decays quadratically, $\epsilon_n(f) = O(n^{-2})$. Furthermore, it is known that this result is optimal in the minimax sense. The best minimax estimator for signals in $\mathbb{V}$ has an approximation error that decays at rate on the same order.

This optimality disappears when one moves into two dimensions. To demonstrate this we need to develop some notation.

Definition 2. *An image f defined on $[0,1]^2$ is in the Horizon Class provided there is a function $H : [0,1] \to \mathbb{R}$ such that $f(x,y) = 1_{\{y \geq H(x)\}}$ $0 \leq x, y \leq 1$.*

The function H is called the horizon of the image. These images are binary, but the results below generalize to arbitrary piecewise constant, horizon model images defined on the unit square.

Definition 3. *For $0 < \alpha \leq 1$ $H \in Holder^{\alpha}(C)$ if*

$$|H(x) - H(x')| \leq C|x - x'|^{\alpha}, \ 0 \leq x, x' \leq 1 \tag{1}$$

For $1 < \alpha \leq 2$ $H \in Holder^{\alpha}(C)$ if

$$|H'(x) - H'(x')| \leq C|x - x'|^{\alpha - 1}, \ 0 \leq x, x' \leq 1 \tag{2}$$

The Holder classes imply a level of fractional regularity on the horizon function. For $\alpha = 1$ we are imposing a Lipschitz condition on H and for $\alpha = 2$ we are imposing a Lipschitz condition on H'.

Definition 4. $Horiz^{\alpha}(C_1, C_{\alpha}) = \{f : H \in Holder^{\alpha}(C_{\alpha}) \cap Holder^{1}(C_1)\}$

We would hope that in improving the Holder regularity of the horizon function we could use the additional information encoded by the smoothness to obtain better nonlinear estimates. It can be shown for images in the class $Horiz^{\alpha}$ that the approximation error for wavelets using a standard tensor product construction to create a basis for $[0,1]^2$ is $\epsilon_n(f) = O(n^{-1})$. However, it can be shown that the optimal minimax estimator is

$$M^*(n, Horiz^{\alpha}) = O(n^{-2\alpha/\alpha+1}).$$

Thus, the standard tensor product wavelet approximation no longer realizes the optimal minimax error estimates. Heuristically the tensor wavelets are interpreting the horizon as a collection of isolated points instead of capturing the 'edginess' as a whole.

2.2. **The Wedgelet Transform.** We proceed as Führ in [3]. Again let f be a piecewise constant function defined on $[0,1]^2$ with piecewise C^2 boundaries between regions. Our objective is to derive an approximating scheme that realizes, up to a constant, the optimal minimax estimate for functions in $Horiz^{\alpha}(C_1, C_{\alpha})$. We begin by partitioning the unit square into dyadic partitions at multiple scales.

Definition 5. $Q_j = \{[2^{-j}k, 2^{-j}(k+1)] \times [2^{-j}l, 2^{-j}(l+1)] : 0 \leq k, l \leq 2^j\}$ *is the set of syadic squares on $[0,1]^2$ at scale j.*

- *Let* $\mathbb{Q} = \bigcup_{j=0}^{\infty} Q_j$.
- *A dyadic partition* Q *of* $[0,1]^2$ *is a tiling of* $[0,1]^2$ *by disjoint dyadic squares.*
- *A wedglet partition* $W = \{(Q_{j,k,l}, \omega_1, \omega_2)\}$ *splits each element of* Q *into at most two wedges* ω_1, ω_2 *along a given line parameterized by an angle* θ *and an offset* k.

Definition 6. *A wedgelet segmentation is a pair* (g, W) *consisting of a wedge partition* W *and a function* g *which is constant on each* $\omega \in W$ [3].
The wedgelet approximation of f *is given by*

$$min_{(g,W)} ||f - g||_2^2 + \lambda |W|$$

The constant value that is chosen for $g|\omega_i$ on a given dyadic square is obviously the average value of the function f on that restricted domain. The function of the wedgelet partition is to capture edges that are not aligned with the coordinate axes at each dyadic scale. Constructing our approximating functions in this way leads us to the following theorem.

Theorem 2.1. *Let* f *be piecewise constant with* C^2 *boundary. Assume that the set* L_j *consists of all lines taking the angles* $\{-\pi/2 + 2^{-j} l\pi : 0 \le l \le 2^j\}$. *Then the nonlinear wedgelet approximation rate for* f *is* $O(|W^{-2}|)$, *i.e., there exists a wedglet segmentation* (g, W) *with* $||f - g||_2^2 \le C|W|^{-2}$.[3]

We will not reproduce the proof here. The basic procedure is to count the number of dyadic squares at scale j that meet the boundary between two constant regions. One then uses Taylor's theorem to approximate the error on each of those dyadic squares where the partitioning wedge line on those squares is a linear approximation to the boundary between the two regions.

We have thus recovered the optimal minimax convergence rate for this class of images. Our justification for using wedgelets to model urban terrain data stems from the fact that urban data is characterized by the presence of strong (usually straight) edges. It was necessary to expand the class of approximating functions from piecewise constant. This allowed us to better approximate structures with sloped roofs.

3. Use of Wedgelets on Digital Elevation Data

3.1. **Data Sets.** The elevation data is generated by a survey aircraft equipped with a LIDAR system. The data returned by the aircraft is called a point cloud.

Definition 7. *A point cloud is an* $n \times 3$ *matrix with coefficients in* $\mathbb{R}$ *where each row represents a point in* $\mathbb{R}^3$.

The data set can be thought of as a discrete random sample of the continuous elevation function.

3.2. **Implementation.** Let $A_{top} = [\vec{x_{top}}, \vec{y_{top}}, \vec{1}]$ contain the (x,y) positions for all of the points in a give dyadic square on one side of a given line. Let $A_{bottom} = [\vec{x_{bottom}}, \vec{y_{bottom}}, \vec{1}]$ contain the same information for the other side of a given line dividing the dyadic square. The optimal approximating function on this dyadic square for this particular line can be defined as

$$g(x,y)|_{\omega_1} = dx_1 x + dy_1 y + z_1$$
$$g(x,y)|_{\omega_2} = dx_2 x + dy_2 y + z_2$$

where $\vec{g_{\omega_1}} = [dx_1, dy_1, z_1]$ is the least squares solution to $A_{top}\vec{g_{\omega_1}} = \vec{z_{top}}$ and $\vec{g_{\omega_2}} = [dx_2, dy_2, z_2]$ is the least squares solution to $A_{bottom}\vec{g_{\omega_2}} = \vec{z_{bottom}}$.

Note that on any wedge, the approximating function g has three degrees of freedom. On a particular dyadic square, the approximating function has eight degrees of freedom.

- Three; dx_1, dy_1, z_1, determine the function $g|_{\omega_1}$.
- Three; dx_2, dy_2, z_2, determine the function $g|_{\omega_2}$.
- Two specify the location of the line in the dyadic square that separates ω_1 and ω_2.

Our algorithm at each step stores the best wedge configuration in terms of L_1 error for each dyadic square in a quad-tree data structure. The L_1 norm was chosen over the L_2 error norm because the L_1 decision tended to pay more respect to discontinuities in that data than the L_2 norm, which is known as a smoothing norm.

The following parameters are user set:

- j_{max}, The smallest scale for the dyadic squares.
- Δk The minimum line translation.
- $|\theta|$ The number of angles used in parameterizing lines.

j_{max} and Δk should be chosen small enough to capture all of the information at the scale of interest.

The parameters Δk and $|\theta|$ serve to parameterize a dictionary of lines for the algorithm to use as wedge partitions. Namely, if the domain in which the x,y coordinates of the point cloud are contained lies in a square $[a, a+c] \times [b, b+c]$, then there are on the order of $\frac{\theta c}{\Delta k}$ lines in the dictionary. The angles used are in a uniform partition of $[-\pi/4, 3\pi/4]$.

The parameter j_{max} serves as a recursion stop.

Pseudo Code for the Wedgelet Transform

For $\theta_1, \theta_2, ...\theta_{|\theta|}$

For $k = k_{min} : \Delta k : k_{max}$

- Split the point cloud matrix into two parts PC_{top} and PC_{bottom}
- Approximate the points in PC_{top} by the best fit least squares regression plane.
- Approximate the points in PC_{bottom} by the best fit least squares regression plane.
- If the sum of the residuals is the smallest compared to all lines checked so far, store this wedge configuration for this dyadic square at the root of a quadtree.
- Dyadically subdivide PC_{top} and PC_{bottom} each into four submatricies and recurse until j_{max} is reached, storing intermediate results in the quad-tree as you move down.

---

k_{min} and k_{max} are determined by the minimal and maximal values for the line offset that cause lines with slope θ to intersect the point cloud domain. PC_{top} and PC_{bottom} refer to points in a particular dyadic square that are above or below a given line.

Generally the QR factorization is used to compute the regression planes. There are a few special cases which deserve comment. If there are fewer than three data points in a given wedge, then the corresponding regression problem is ill defined. As a preprocessing step, we preform regression on each dyadic square with no line passing through it. Then when the situation occurs in which either PC_{top} or PC_{bottom} is empty, we do not need to do anything. If either PC_{top} or PC_{bottom} contains one point, then the plane is set to be a constant taking the value of that point. If either matrix contains two points, then we simply average.

Also, if either matrix contains fewer than 20 points, then we use SVD instead of QR to find the best fit plane. We do this because as the number of points shrinks, the probability of the points contained in the matrix become linearly dependent increases. SVD allows us to zero out singular values that are close to zero and solve the regression equations in terms of the other values. This helps ensure stability of the algorithm.

4. Results

4.1. **Sample Data Sets.** To test wedgelets on real LIDAR data we used two ready made DEMs. One was taken over Ft. Belvior, VA. The other was taken over New Orleans, LA.

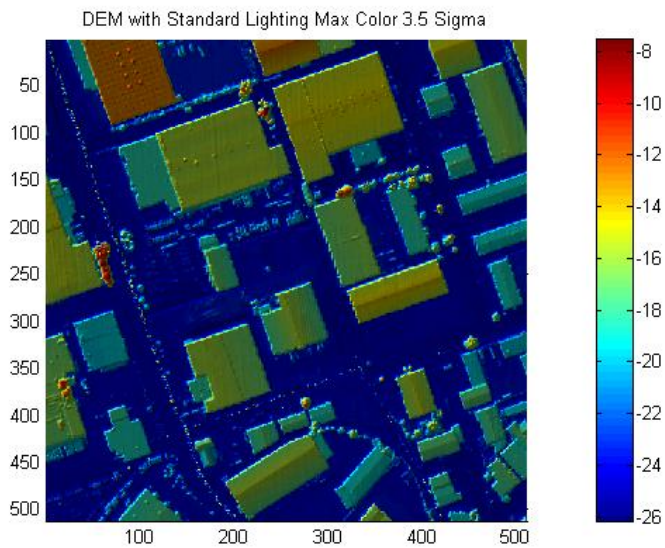

Figure 1. New Orleans

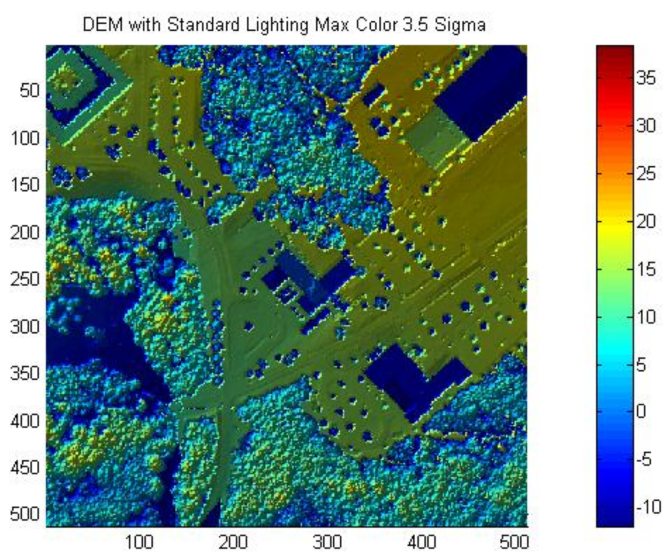

Figure 2. Ft. Belvor

The New Orleans data set was our canonical urban data set while the Ft. Belvior data set was our canonical non urban data set. To generate point clouds from this data, we randomly sampled the domain and assigned values to the points using bilinear interpolation. We sampled so that our point cloud has an average density of 3 $data\ points/m^2$.

4.2. **Comparing Images: Choosing a Quality Metric.** In order to assess the quality of the DEMs generated from the wedgelet transform, we compared them to the reference DEMs provided by the Army Corps of Engineers. The measure of quality used is a variation of the Sturctural Similarity Index described in [6]. This is a full reference measure. If X is an image recovered using wedgelets and Y is the reference image, the TSSIM measure has the following properties:

- Symmetry: $TSSSIM(X,Y) = TSSIM(Y,X)$
- Boundedness: $TSSIM(X,Y) \leq 1$
- Unique Maximum: $TSSIM(X,Y) = 1 \iff X = Y$

TSSIM is computed by computing image statistics locally and then computing a similarity measure for both the contrast and image structure.

$$c(X_{local}, Y_{local}) = \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad s(X_{local}, Y_{local}) = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

These indivudal measurements both satisfy the properties of TSSIM listed in the previous frame. The global TSSIM measure is computed by averaging the products of $c(X,Y), s(X,Y)$ over all of the local observations. Empirical observation has shown that a TSSIM at or above 0.75 indicates a very strong similarity between two images.

Observation has shown TSSIM to be an excellent indicator of objective image quality. It is better suited to the task of assessing the qualities of images than the standard mathematical error norms, as it pays more attention to local structures.

4.3. **Compression Rates.** The reference models are 512^2 pixels. We determine the compression rate by comparing the number of doubles that need to be stored to create a given wedgelet approximation to 512^2. The results are shown below.

For the urban DEM we were able to reconstruct to an acceptable level of quality at 85% compression. The rural data fared much worse. This is intuitive, however, as the trees in the Ft. Belvior data will not yield themselves to approximation by planes. The Ft. Belvior data is too far from the class of images that wedgelets were intended for.

It should also be noted that increasing the angular resolution increased the quality of approximation only slightly.

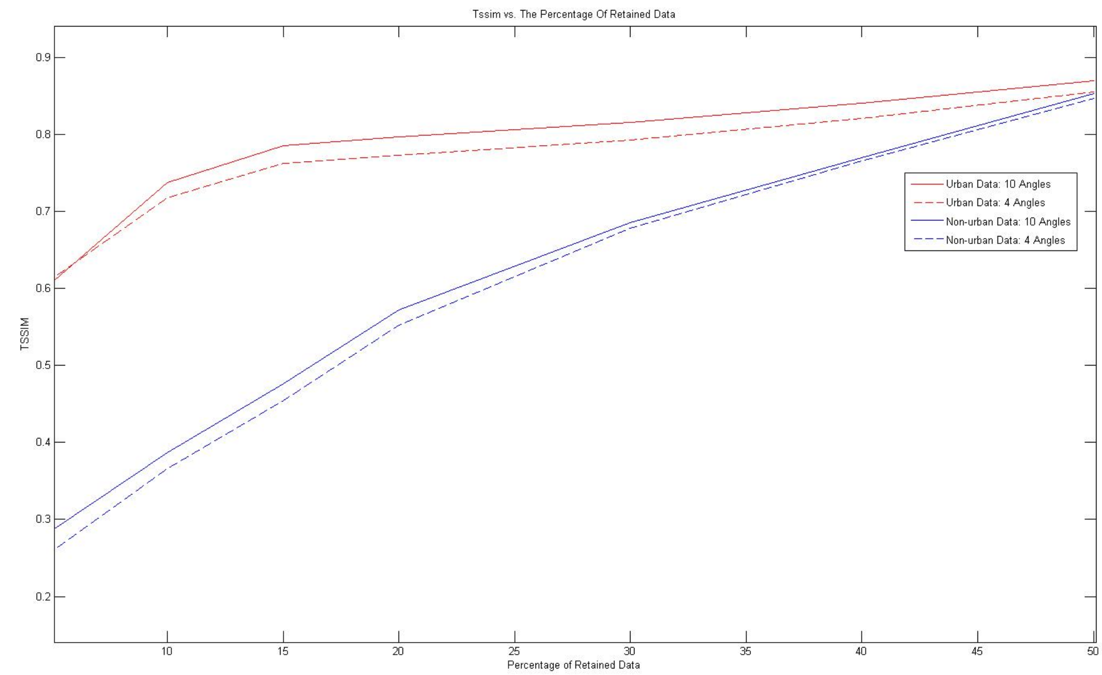

FIGURE 3. Percentage of Retained Data vs TSSIM

4.4. **Wedgelet Denoising.** Using wedgelets to denoise images containing gaussian noise is described in [2]. The basic idea is to find a sweet spot where the wedges you use are large enough so that the interpolative process removes the noise but small enough that detail is preserved. We took our the urban reference point cloud and added gaussian white noise with zero mean and $\sigma = \frac{1}{3}$. Resulting wedgelet reconstructions are shown below.

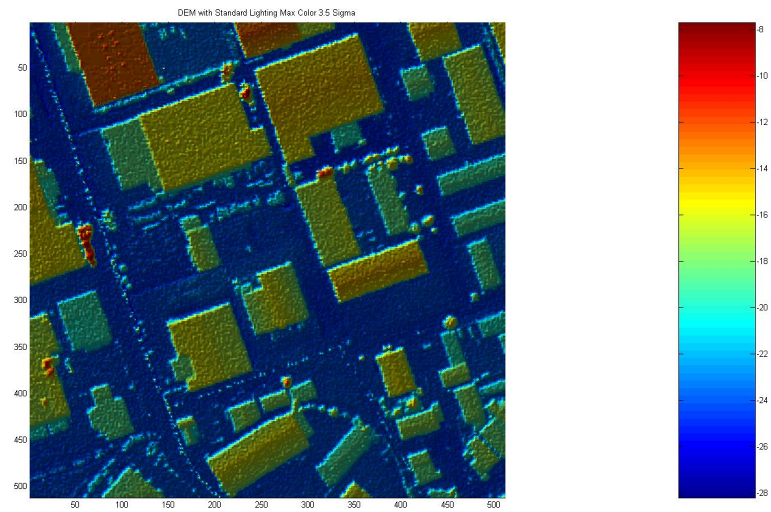

FIGURE 5. 30% retained data

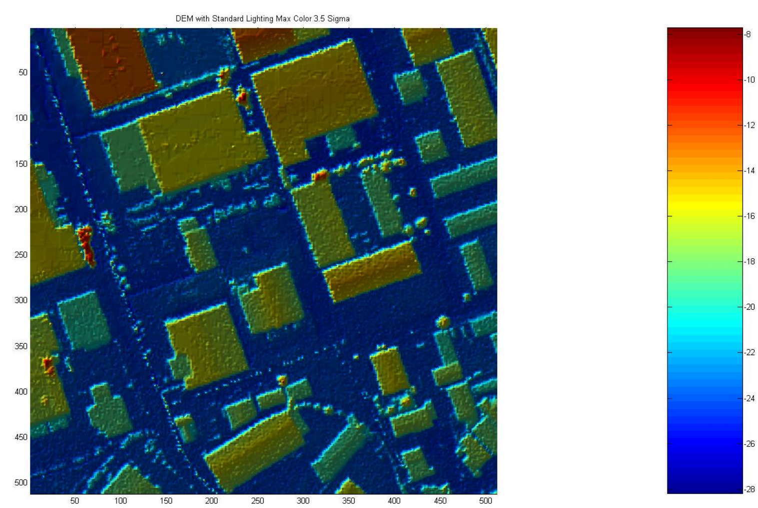

FIGURE 4. 15% retained data

Note the sudden disappearance of noise between the 30% and 15% compression levels. A plot of percentage of retained data vs TSSIM illustrates this well.

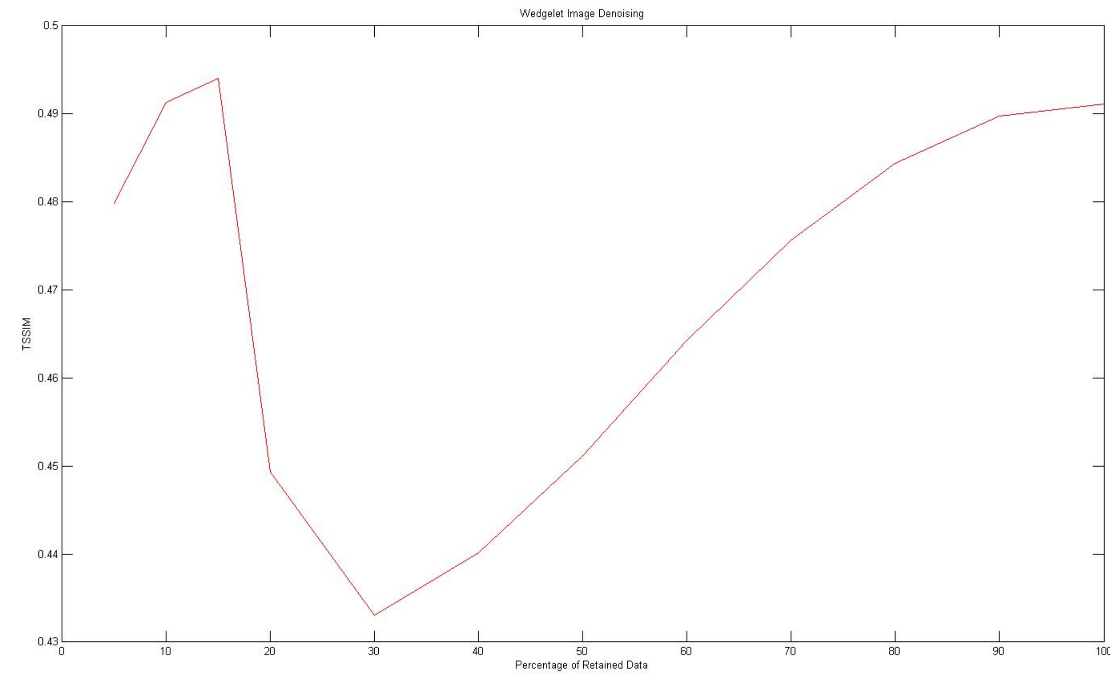

Figure 6. Percentage of Retained Data vs. TSSIM For noisy data

It is obvious that increasing the sampling density of the point cloud and/or reducing the level of noise will move the peak in the above plot toward the right. This indicates that de-noising could be done in these cases with a less significant loss of image detail. We do not quantify these observations here.

We also examined the case where instead of perturbing the elevation portion of the data, the (x, y) information was perturbed. We took each point in our urban reference point cloud and added a gaussian random variable with mean zero and $\sigma = \frac{1}{15}$ to both the x and y coordinate. The value for σ was chosen to be similar to realistic sensor noise associated with a GPS system. The reconstruction was preformed using a very small level of compression ($\approx 5\%$ *compression*).

Under these circumstances wedgelets performed very well. The interesting observation is that on the interior of buildings, the wedgelet reconstruction was essentially perfect. When the perturbation 'kicks' points over an existing edge, there is a small yet measurable distortion. This manifests as edge blurring. Figure seven is a plot of the absolute value of the difference between the reference DEM and the reconstruction with noisy data. The deep blue regions are within singular precision of zero. This demonstrates that wedgelets is having no problem coping with this type of noise on the interior of bounded regions.

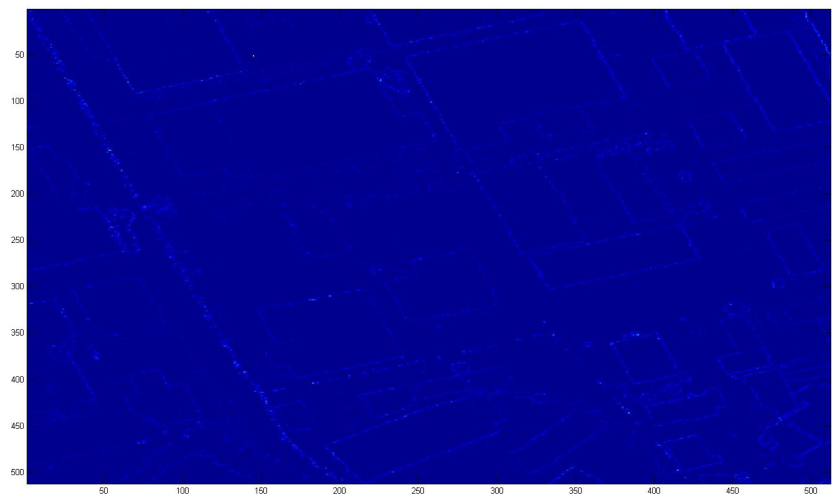

FIGURE 7. Difference Between Reference Image and Reconstruction From Noisy Data

REFERENCES

[1] David L. Donoho *Wedgelets: Nearly Minimax Estimation of Edges*, The Annals of Statistics, Vol. 27, No. 3. (Jun., 1999), pp. 859-897.

[2] Laurent Demaret, Felix Friedrich, Hartmut Fhr, Tomasz Szygowski, *Multiscale Wedgelet Denoising Algorithms*, Proceedings of SPIE, San Diego, August 2005, Wavelets XI, Vol. 5914, X1-12

[3] Beyond Wavelets: New image representation paradigms H.Führ, L.Demaret, F.Friedrich Survey article, in Document and Image Compression, M.Barni and F.Bartolini (eds), May 2006

[4] Moenning C., Dodgson N. A.: *A New Point Cloud Simplification Algorithm.* Proceedings 3rd IASTED Conference on Visualization, Imaging and Image Processing (2003).

[5] S. Sotoodeh , *Outlier Detection In Laser Scanner Point Clouds*, IAPRS Volume XXXVI, Part 5, Dresden 25-27 September 2006

[6] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.