Maja Kucharczyk
2018-04-23
Geog 639
Term Project Final Report

Logistic regression to predict roof damage using GLCM texture from post-hurricane UAV imagery

1. Introduction

1.1 Background

Natural disasters such as earthquakes and tropical cyclones cause vast amounts of death and destruction worldwide. In the wake of these events, timely earth observation data can help coordinate response efforts by identifying collapsed buildings, inaccessible roads, and more (Plank, 2014). Building damage assessments (BDAs) by remote sensing methods are important for the response phase (rapid damage mapping for search and rescue operations), as well as for the later and longer-term recovery phase (economic loss assessment and reconstruction) (Dong and Shan, 2013; Fernandez Galarreta et al., 2015; Vetrivel et al., 2015). Information relevant to disaster management includes extent and amount of damage, number of collapsed buildings, grade of building damage, and type of building damage (Dong and Shan, 2013).

Generally, building-level damage can only be detected with remote sensing data with a spatial resolution of 1 m or finer (ideally 0.5 m or finer), whereas coarser resolutions can only be used for detecting block-level damage (many buildings grouped together) (Dong and Shan, 2013). The types of remote sensing data used for building damage assessment are optical imagery, synthetic aperture radar, and light detection and ranging (Dong and Shan, 2013). Methods can either be mono-temporal, where damage is detected using post-event data only, or multi-temporal, where pre- and post-event data are used (Dell'Acqua and Gamba, 2012; Dong and Shan, 2013).

In the response phase of disaster management, where time is essential, satellite data acquisition, processing, and delivery can experience several bottlenecks (Ajmar et al., 2015; Denis et al., 2016). For example, based on 41 activations of the International Charter for Space and Major Disasters in 2014, the average response time (of emergency mapping delivery) was 2.9 days, exceeding the 24-

hour requirement for the information to "contribute efficiently and effectively to the management of operations" (Denis et al., 2016). While improvements are being made to the timeliness of emergency optical and radar satellite data products, unmanned/uninhabited aerial vehicles (UAVs) have been speculated as potential sources of rapid, actionable information for first responders (Ajmar et al., 2015). The majority of commercially available, lightweight UAVs carry consumer-grade RGB cameras. The images captured are processed using structure from motion (SfM) photogrammetric software to yield a high resolution orthorectified mosaic. An orthomosaic generated from post-event imagery can be used for mono-temporal building damage assessment.

1.2 Objectives and hypothesis

The primary objective of this project was to build a logistic regression model to predict the presence or absence of roof damage using a single band of a UAV RGB orthomosaic. This was achieved by first conducting a variogram analysis of sample damaged buildings in order to observe the spatial autocorrelation of pixel values. The single band image was then used to extract grey level co-occurrence (GLCM) texture measures from buildings with intact roofs and buildings with damaged/missing roofs. Since buildings with damaged/missing roofs appear to have greater contrast and disorder of pixel values than those with intact roofs, it was hypothesized that the GLCM textures pertaining to contrast and orderliness would be strong predictors of roof damage. The logistic regression models were assessed for classification accuracy using a separate testing dataset, and compared against Random Forest classification models.

2. Literature Review

2.1 Mono-temporal methods for building damage detection using post-event optical imagery

The main methods for building damage detection using post-event satellite and airborne optical imagery have been manual visual interpretation, as well as the use of spectral, morphological, and textural information for more automation (Dong and Shan, 2013). Visual interpretation of satellite imagery is the method that appears to be the most common and operational on a large scale. For example, Copernicus Emergency Management Service (CEMS) employs visual interpretation of post-event satellite imagery to rapidly produce damage grading maps within hours or days following a disaster (CEMS, 2018). Their damage grading maps include information such as damage extent, number of damaged buildings per cell of a grid, and grade of damage per building (CEMS, 2018).

2.2 The role of UAVs in building damage detection

UAVs have been explored for their usefulness in conducting response-phase rapid mapping to detect building damage. Chen et al. (2016) used UAV imagery captured after an earthquake to test three major types of texture features for classifying building damage on a five-degree scale. The texture feature types were GLCM, Tamura, and Gabor wavelet (Chen et al., 2016). The texture-derived damage classifications were assessed with a Correct Classification Ratio (CCR) metric (Chen et al., 2016). The CCR was 64% when using GLCM texture measures only, 73% when using GLCM and Tamura texture measures, and 88% when using GLCM, Tamura, and Gabor texture measures (Chen et al., 2016). However, this study did not provide details such as UAV image properties, data acquisition properties, building sample size and properties, and the source of the reference data used to evaluate classification accuracy. Sui et al. (2014) also used GLCM texture measures from UAV imagery to detect building damage; however, their method used pre- and post-earthquake UAV imagery for change detection.

Rapidly acquired UAV imagery may be used to augment services like CEMS. CEMS provided damage grading of over 13,000 buildings in Sint Maarten 2-4 days after the service activation. The first damage grading map was released in 2 days, but the satellite image used had 50% cloud cover in the area of interest (CEMS, 2017). An updated, second damage grading map was released 2 days later, when satellite imagery with lower cloud cover (15%) became available (CEMS, 2017). In this type of scenario, if UAVs and operating teams are local, UAV imagery could be acquired more quickly than satellite imagery, and could be used to provide information to first responders more quickly than in 2-4 days. The high resolution orthomosaic can be visually interpreted, but semi-automated and automated methods of classification should also be explored. Chen et al. (2016) showed that GLCM textures, as well as other texture measures, can be successful in training building damage classifiers. This project will focus on using GLCM textures to train models for detecting missing/damaged roofs.

2.3 GLCM textures

Texture analysis on imagery can reveal contrast (roughness), the neighborhood over which a change occurs (coarseness), and the directional properties of the changes (directionality) (Hall-Beyer, 2017a). Texture measures often improve the accuracy of image classification by adding object (e.g., land cover) properties that are complimentary to spectral measures (Hall-Beyer, 2017a). The most commonly used texture measures are those of the grey level co-occurrence matrix (GLCM) (Hall-Beyer, 2017a). The simple definition of a GLCM is a matrix that shows how often different two-pixel combinations of brightness levels (grey levels) occur in an image (Hall-Beyer, 2017a). This “co-occurrence” of two pixel values can be defined as immediate neighboring (one pixel away) or any x and y offset (horizontal, vertical, or diagonal) from the “reference” pixel to the “neighbor” pixel (Hall-Beyer, 2017a).

The GLCM is used to derive second-order texture measures that describe spatial relationships between two pixel values (Hall-Beyer, 2017a). The GLCM is calculated and used to derive texture values for each pixel in an image using a moving window (Hall-Beyer, 2017a). Haralick et al. (1973) and Haralick (1979) suggested 14 GLCM-based texture measures. There are 8 that are commonly implemented in software (Hall-Beyer, 2017a). They can be organized into 3 groups (with the measures in parentheses): (i) Contrast Group (contrast, dissimilarity, homogeneity), (ii) Orderliness Group (angular second moment, entropy), and (iii) Descriptive Statistics of the GLCM Group (mean, variance, correlation) (Hall-Beyer, 2017). GLCM standard deviation is sometimes calculated instead of GLCM variance (Hall-Beyer, 2017a). The equations for these 8 measures are:

$$Contrast = \sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2$$

$$Dissimilarity = \sum_{i,j=0}^{N-1} P_{i,j}|i-j|$$

$$Homogeneity = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2}$$

$$Angular\ Second\ Moment\ (ASM) = \sum_{i,j=0}^{N-1} {P_{i,j}}^2$$

$$Entropy = \sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j})$$

$$GLCM\ Mean = \mu_i = \sum_{i,j=0}^{N-1} i(P_{i,j}) \quad ; \quad \mu_j = \sum_{i,j=0}^{N-1} j(P_{i,j})$$

$$GLCM\ Standard\ Deviation = \sigma_i = \sqrt{\sum_{i,j=0}^{N-1} P_{i,j}(i-\mu_i)^2} \quad ; \quad \sigma_j = \sqrt{\sum_{i,j=0}^{N-1} P_{i,j}(j-\mu_j)^2}$$

$$GLCM\ Correlation = \sum_{i,j=0}^{N-1} P_{i,j} \left[\frac{(i - \mu_i)(j - \mu_j)}{(\sigma_i)(\sigma_j)} \right]$$

where $P_{i,j}$ is the probability value at row i (reference pixel value) and column j (neighbor pixel value) of the GLCM (Hall-Beyer, 2017a).

Liu and Liew (2007) extracted GLCM texture measures from pre- and post-earthquake Ikonos panchromatic images. They calculated the mean value for four GLCM texture measures (contrast, entropy, homogeneity, energy) for 25 sample patches over buildings in both images (Liu and Liew, 2007). The mean values for the GLCM textures showed that the damaged buildings had higher contrast and disorder than the undamaged buildings (Liu and Liew, 2007).

Hall-Beyer (2017b) found that the common 8 GLCM texture measures can be grouped by those useful for edge detection, and those useful for discriminating between different land cover patches. Contrast, dissimilarity, entropy, and GLCM variance are useful for highlighting edges of land cover patches, but are not useful in discriminating between land covers (Hall-Beyer, 2017b). Homogeneity, GLCM mean, GLCM correlation, and angular second moment (ASM) are useful for describing the interior of land cover patches – GLCM mean and GLCM correlation have been found to be most useful for discriminating between different land covers (Hall-Beyer, 2017b). This information may be useful when using GLCM textures to discriminate between damaged and undamaged buildings. Though they did not explain why, Chen et al. (2016) used ASM, GLCM correlation, and entropy to classify building damage using UAV imagery.

2.4 Logistic regression

This project will use logistic regression to evaluate the use of GLCM textures in predicting the presence or absence of building damage. Logistic regression is appropriate since the dependent variable is binomial, having observed values of either 0 (undamaged) or 1 (damaged) (Geldsetzer, 2018a). This type of regression is non-linear; the logistic curve ensures that the predicted values will be within the range of possible values (Rogerson, 2006). The equation of the logistic regression curve is:

$$\hat{Y}_i = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+ \ldots \beta_N X_N}}{1 + e^{\beta_0+\beta_1 X_1+\beta_2 X_2+ \ldots \beta_N X_N}}$$

where $\hat{Y}_i$ is the estimated probability of being in one binary outcome category i versus the other, β_0 is the y-intercept, β_1 is the beta coefficient of the first independent variable, X_1 is the value of the first independent variable, and N is the total amount of independent variables (Stoltzfus, 2011). The values predicted by this equation will follow a non-linear trend, but can be transformed into a linear function with the independent variable(s) (Rogerson, 2006). This logistic transform uses the following equation:

$$\ln\left(\frac{\hat{Y}}{1-\hat{Y}}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_N X_N$$

where $\frac{\hat{Y}}{1-\hat{Y}}$ is the conditional probability of being in one outcome category versus the other category (Stoltzfus, 2011). The log of this conditional probability is called the "log-odds" (Rogerson, 2006; Geldsetzer, 2018a). Using this linear function, logistic regression finds the strongest combination of independent variables that maximizes the likelihood of predicting the observed values – this is done with a maximum likelihood estimation (Stoltzfus, 2011).

The assumptions of logistic regression are that: (i) the conditional probabilities are related to the independent variable(s) through a logistic function, (ii) all important independent variables are included in the model, (iii) no unimportant independent variables are included, (iv) there are minimal

measurement errors, (v) the observations are independent, and residuals are not autocorrelated, (vi) multicollinearity is absent, and (vii) strong outliers are absent (Geldsetzer, 2018a).

Because logistic regression uses maximum likelihood estimation, it generally requires more samples than independent variables in order to cycle through different solutions to find the best model fit (Stoltzfus, 2011). Specifically, per independent variable, there should be at least 10 samples of the less-frequent outcome (Stoltzfus, 2011). For example, if the less-frequent outcome has 30 samples, no more than three independent variables should be included in the logistic regression. Regarding independent variable selection, it's advised to perform individual logistic regression with each candidate variable to check for the significance of its contribution (Geldsetzer, 2018a). Multicollinearity should be checked by constructing a correlation matrix, observing which independent variables are correlated with other independent variables, and removing redundant variables (Stoltzfus, 2011). With the appropriate number of remaining, uncorrelated, significant variables, model building can begin. A popular model-building technique is the forward or backward stepwise approach (Stoltzfus, 2011). Forward stepwise regression constructs a model with most significant independent variable first, then adds one variable at a time, and stops when no additional variables are significant (Stoltzfus, 2011). Backward stepwise regression constructs a model with all independent variables first, then eliminates one non-significant variable at a time until all remaining variables are significant (Stoltzfus, 2011).

The goodness-of-fit of the final logistic regression model can be assessed using the Pearson chi-square and residual deviance statistics, both of which are measures of the model residuals (Stoltzfus, 2011). Beyond goodness-of-fit, the user can evaluate the model's performance in discriminating between the two possible outcomes (Stoltzfus, 2011). One method is to predetermine a cut-off point (i.e., a conditional probability value that serves as a threshold for outcome classification), and then produce a classification table (Stoltzfus, 2011). The other method is to use the area under the receiver

operating characteristic curve (AUROC) value, which can range from 0.5 (indicating the model is no better than random chance) to 1.0 (indicating the model perfectly discriminates between the two possible outcomes) (Stoltzfus, 2011).

When assessing independent variables, beta coefficients and odds ratios (ORs) are used (Stoltzfus, 2011). If multiple independent variables were used in the logistic regression model, then the beta coefficient for each independent variable represents the change in the log-odds for a one-unit increase in the independent variable, holding the other independent variables constant (Geldsetzer, 2018a). An OR indicates the strength of an independent variable's contribution to the predicted outcome, expressed as $OR = e^{\beta_i}$ (Stoltzfus, 2011). Logistic regression models with multiple independent variables use an adjusted OR for each independent variable, such that the effects of the other independent variables are removed (Stoltzfus, 2011). For example, an OR of 1.5 would indicate that, for every one-unit increase in the independent variable, the odds of the outcome increase by 50% (i.e., 1.5-1.0=0.5) (Stoltzfus, 2011). Commonly, 95% confidence intervals are reported for the ORs; if the interval contains values less than 1, the significant contribution of the independent variable should be questioned (Stoltzfus, 2011).

2.5 Random Forest classification

Random Forest (RF) (Breiman, 2001) is a machine learning algorithm that has become popular for classifying remote sensing imagery (Millard and Richardson, 2015). RF works by creating Classification and Regression (CART)-like trees, with each tree containing a randomly selected 2/3 subset of the training data (Millard and Richardson, 2015). Each tree (classifier) is used to make a prediction on a testing sample, and all the classifier predictions are considered in the final classification of the sample (Akar and Gungor, 2015). Best practices for RF classification include: (i) using only important,

uncorrelated variables, (ii) performing iterative classifications to assess the stability of predictions, (iii) using as many training and testing samples as possible, (iv) having an unbiased sampling strategy, and (v) minimizing spatial autocorrelation within the training and testing samples (Millard and Richardson, 2015).

3. Methods

3.1 Study area and UAV data

On September 7, 2017, Category 5 Hurricane Irma struck the Caribbean island of Saint Martin (France)/Sint Maarten (the Netherlands) (Figure 1). From September 11 to October 3, a Canadian organization called RescUAV (a program within GlobalMedic) collected UAV imagery in Sint Maarten, covering over 20 km^2 (Figure 2). Thirty six imagery sets were processed into RGB orthomosaics. The orthomosaic covering Great Bay Beach in Philipsburg was chosen for this project (Figure 3). The UAV images were collected on September 12, covering 0.5 km^2 (Figure 3), and processed using Pix4D structure from motion photogrammetric software to produce a 0.03 m resolution orthomosaic.

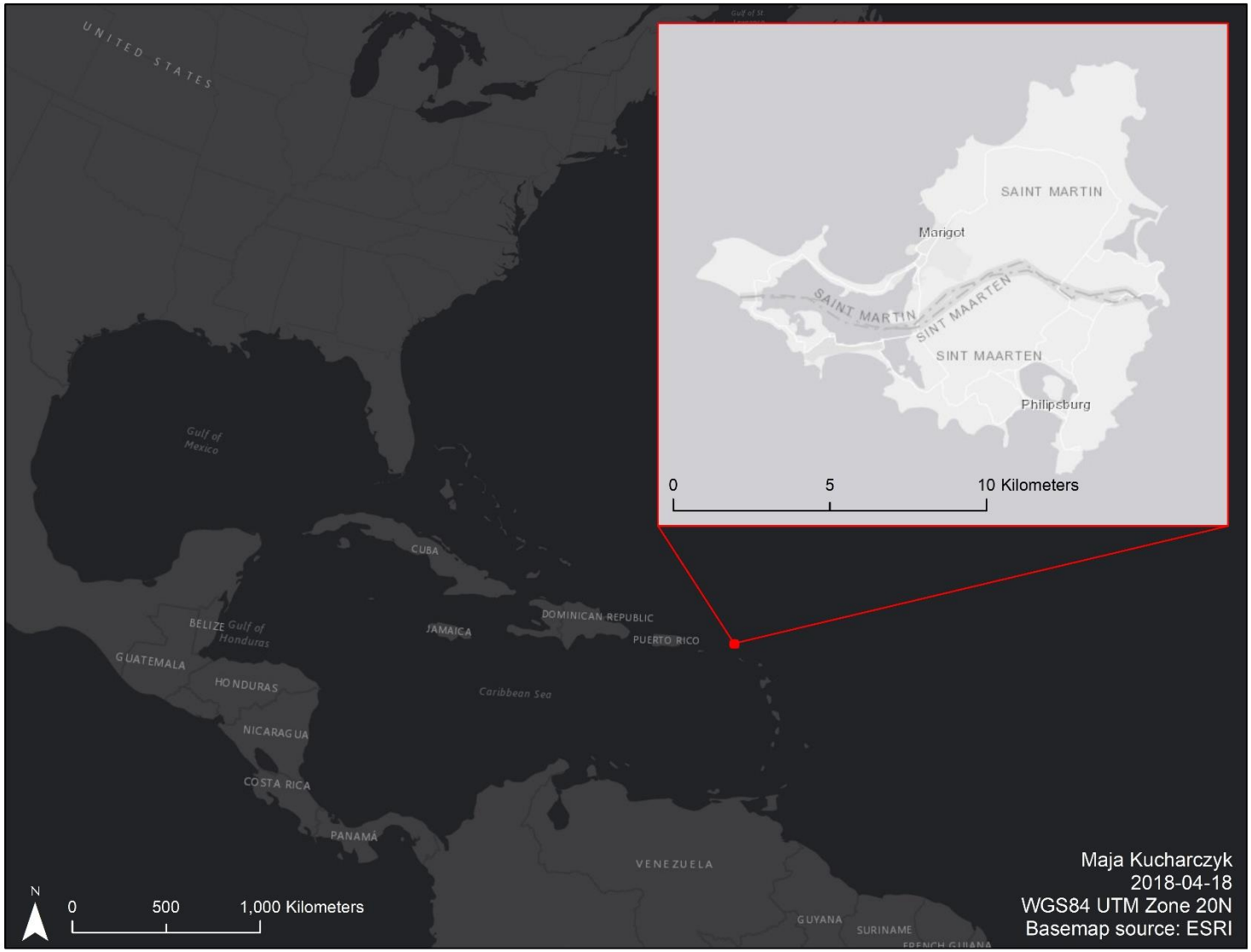


Figure 1. Location of Saint Martin (northern half, French), and Sint Maarten (southern half, Dutch).

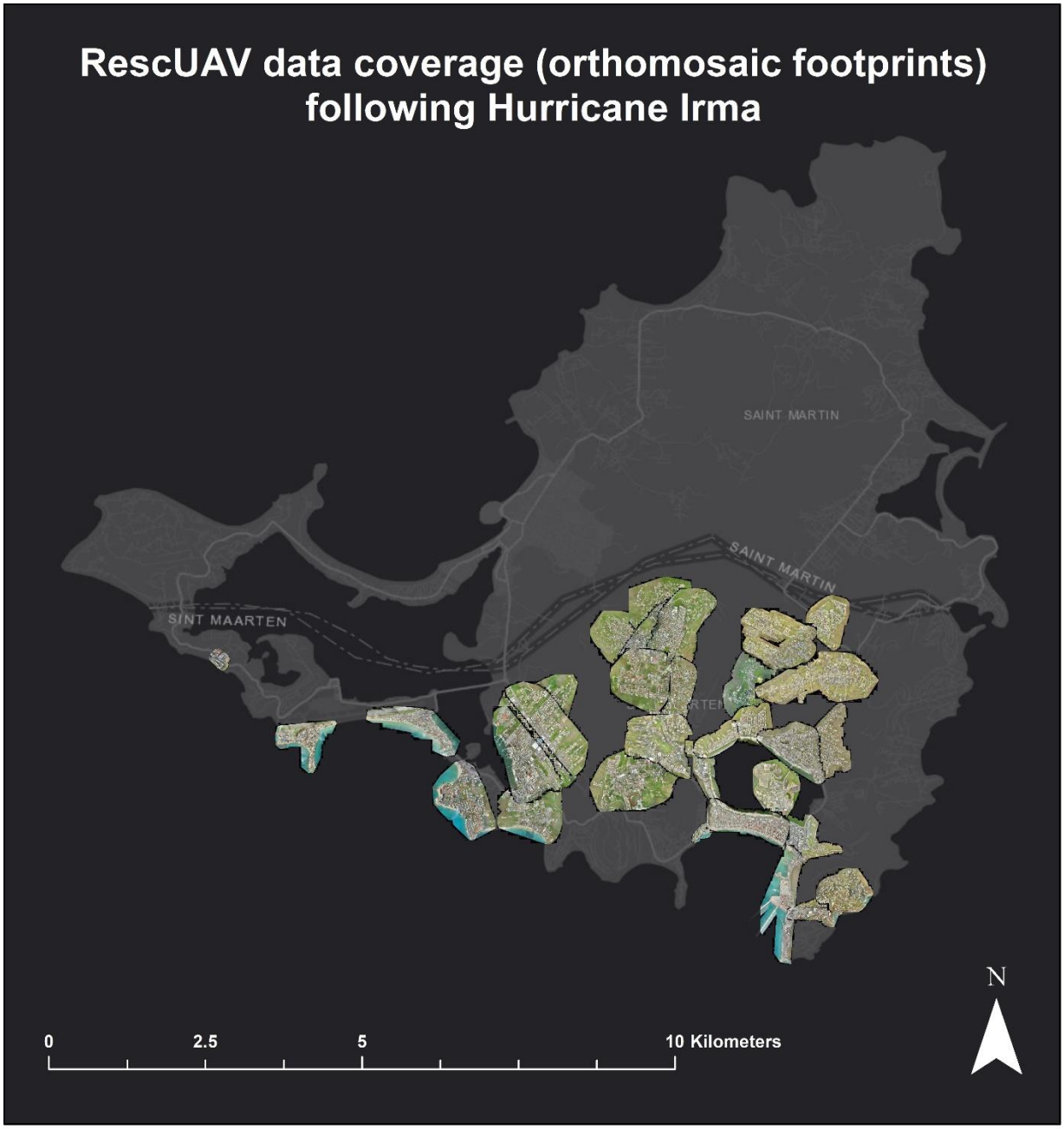


Figure 2. RescUAV UAV data coverage in Sint Maarten.

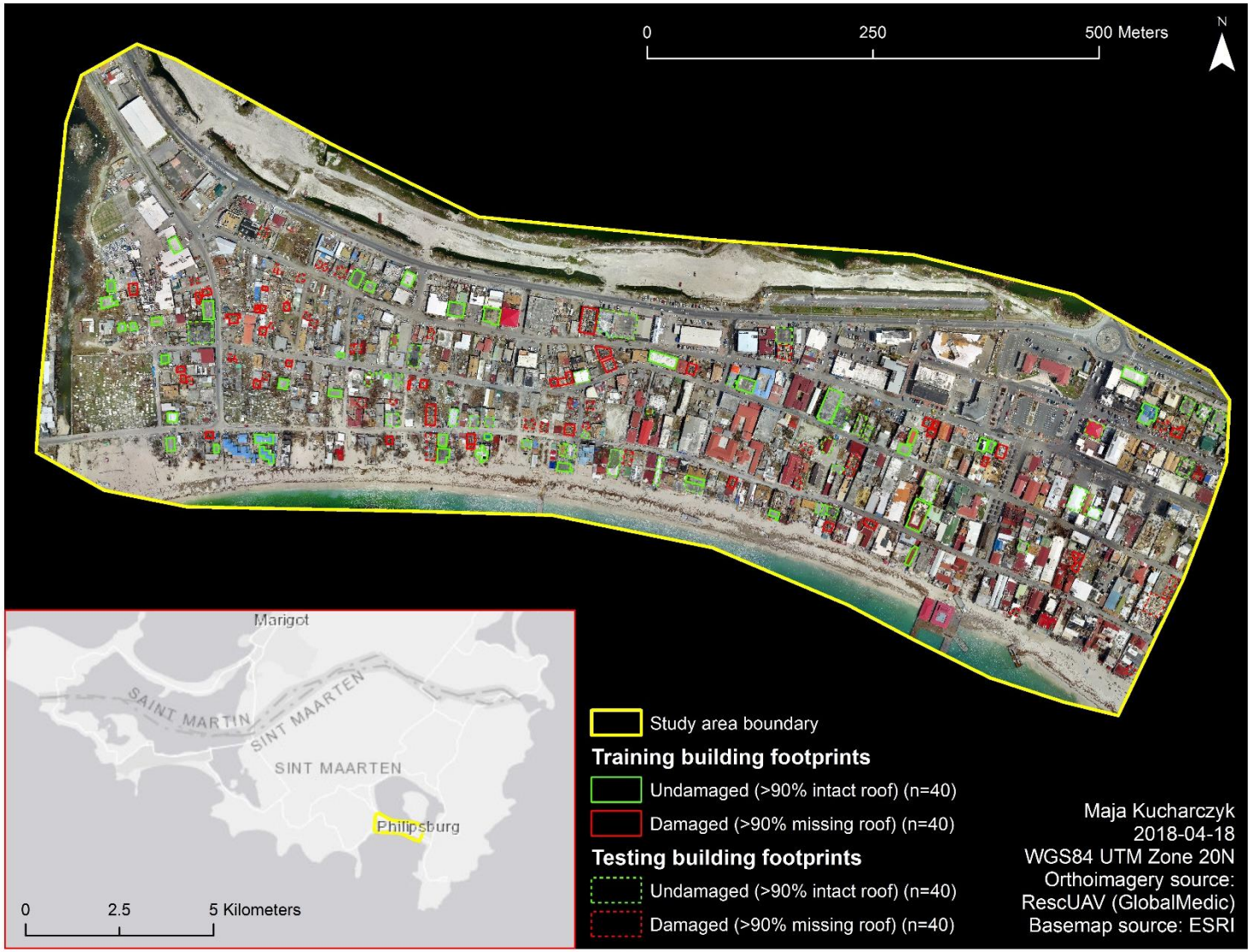


Figure 3. Study area location (Great Bay Beach, Philipsburg, Sint Maarten) and building footprints of training and testing samples.

3.2 Training and testing building footprints

The orthomosaic was used to digitize 160 building footprints. From the 160, 80 buildings had roofs that were >90% intact, and were assigned a value of 0 (undamaged) (Figure 3). Of these 80, 40 randomly selected samples were allocated to the training set, and the other 40 were allocated to the testing set. The remaining 80 buildings had roofs that were >90% damaged/missing, and were assigned a value of 1 (damaged) (Figure 3). Of these 80, 40 randomly selected samples were allocated to the training set, and the other 40 were allocated to the testing set. All spatial datasets were projected in UTM Zone 20N with the WGS84 datum. To provide examples of the range of roof appearances in the damaged and undamaged classes, 10 sample buildings were extracted from each class. The red band of the RGB orthomosaic and sample footprints are shown in Figures 4 and 5. The damaged samples show at least 90% of roof missing (Figure 4). Some damaged samples (e.g.., #3, #4, #5, #8, #9, and #10) show wooden beams in varying configurations, while others (e.g., #1) show no wooden beams (Figure 4). Damaged sample #7 appears to contain collapsed roof material inside the building (Figure 4). The undamaged samples show >90% roof intact (Figure 5). They vary in texture, with #13 appearing as a smooth surface, and the others having rougher textures due to factors such as roof material, roof objects (e.g., #12), and the possible presence of water damage (e.g., #14, #19) (Figure 5).

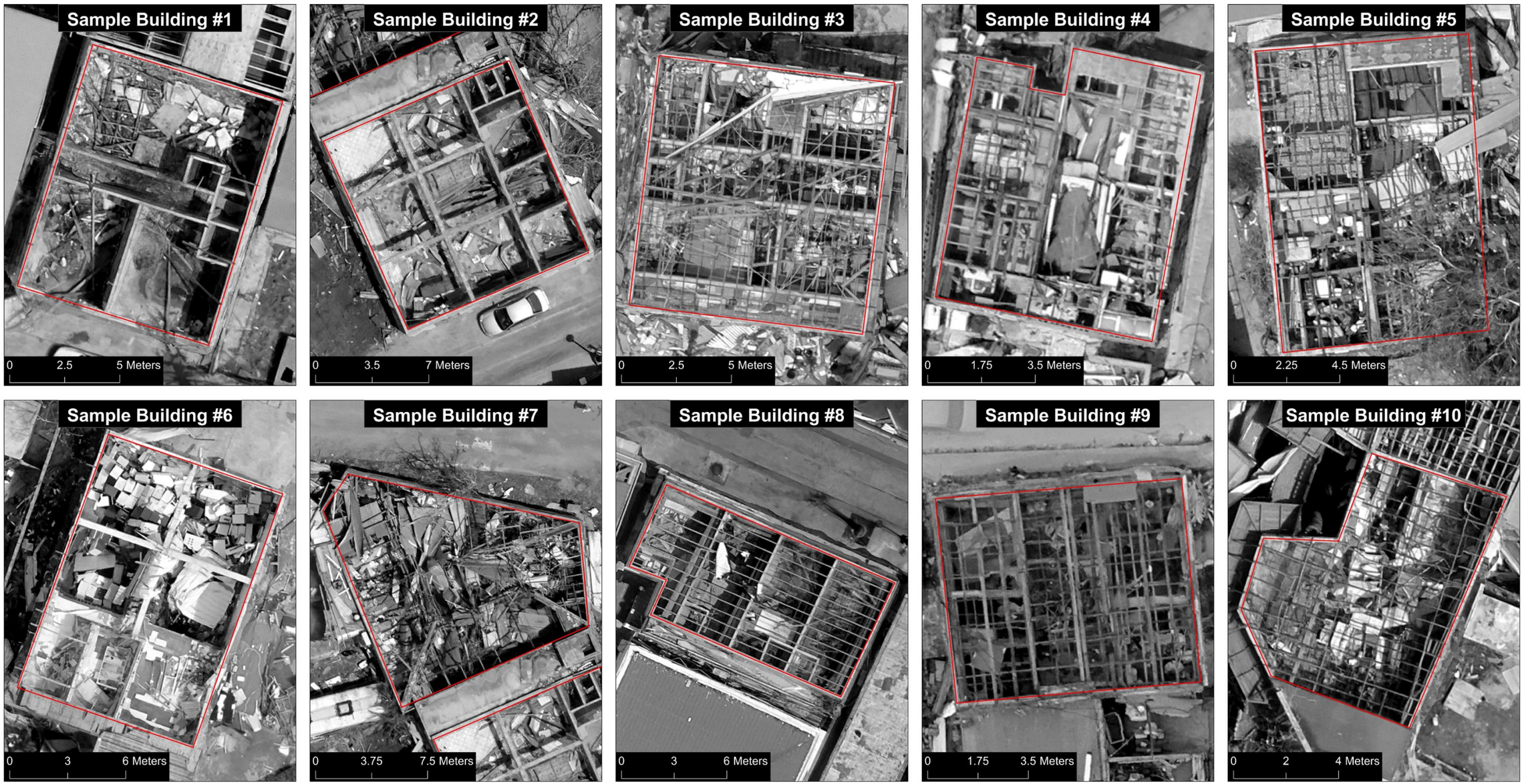


Figure 4. Sample damaged building footprints (red) with UAV orthomosaic red band image.

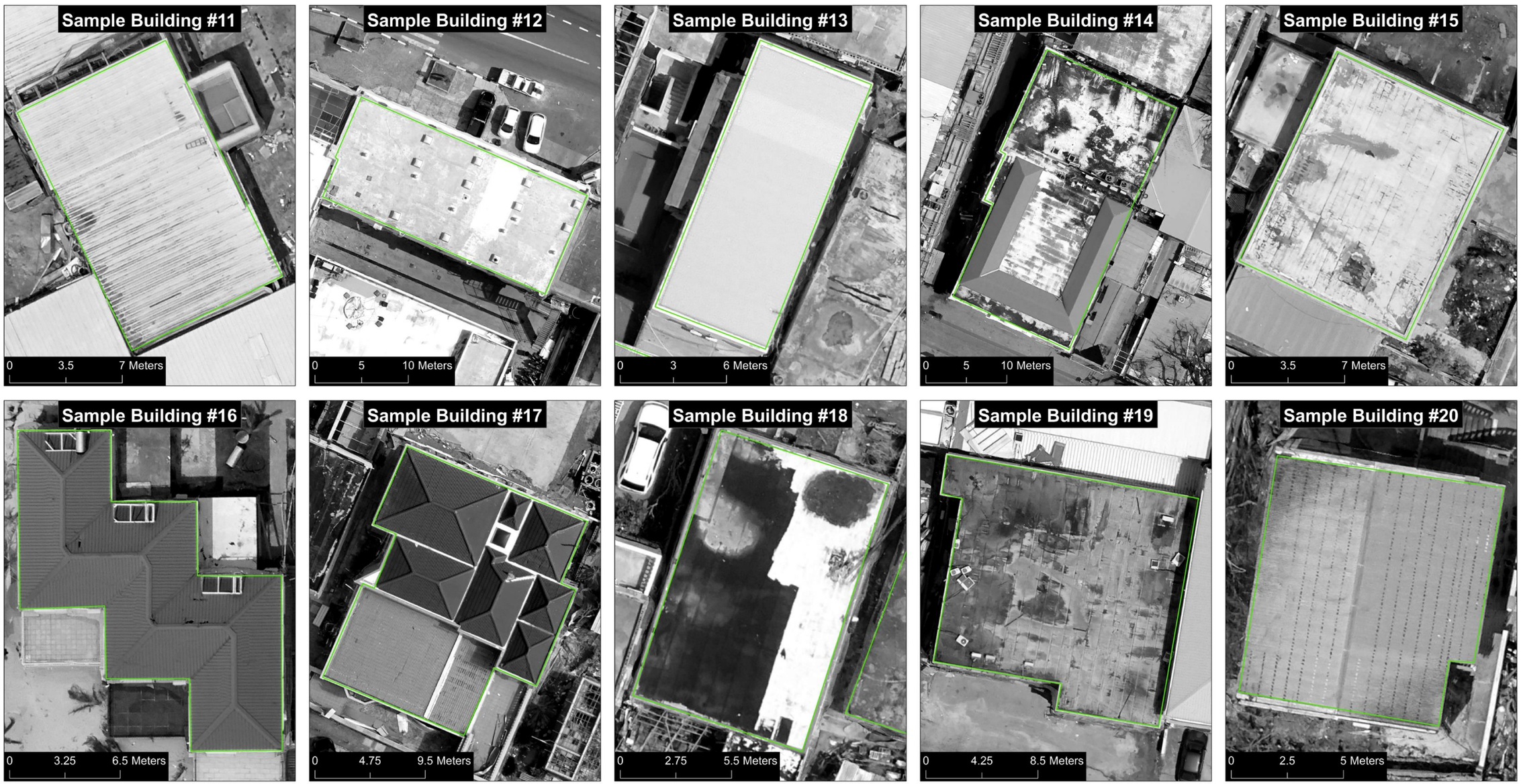


Figure 5. Sample undamaged building footprints (green) with UAV orthomosaic red band image.

3.3 Variogram analysis

The red band of the orthomosaic was selected for the texture analysis, as it had higher contrast than the green and blue band images (Hall-Beyer, 2017b). Before texture image generation, variogram analysis was performed. The first objective of the variogram analysis was to determine which window size to use for the texture analysis. Hall-Beyer (2017a; 2017b) recommends using the variogram range as a guide, as this is the distance over which spatial autocorrelation occurs. To capture the image texture within the window, it is logical to set the window size larger than the variogram range. However, setting the window size excessively large will unnecessarily increase the time required to generate the texture image, as each pixel's texture calculation would require consideration of more pixel pairs. The second objective of the variogram analysis was to determine if there was a maximum pixel size suitable for texture-based damage classification. Due to the very high spatial resolution of the UAV imagery (0.03 m), it is important to determine if a range of pixel sizes (starting from 0.03 m) provides redundant information. The identification of a maximum pixel size suitable for this application has important implications for imaging platforms and, therefore, optimal data areal coverage in a rapid response context.

For the variogram analysis, a variogram model was constructed for each of the 10 damaged samples shown in Figure 4. To do this, a series of experimental variograms were first generated in order to find suitable variogram parameters for the final models. The red band image was clipped by the footprints of the 10 damaged samples. For each damaged sample, the clipped image was used to generate an experimental variogram using R (v. 3.4.3) via RStudio (v. 1.1.383) (R Core Team, 2017), using the package "usdm". A recommended method for choosing the maximum lag distance is use 1/3 to 2/3 of the maximum distance of the data extent (Geldsetzer, 2018b). In this case, the maximum diagonal lengths of the 10 sample buildings ranged from 11.2 to 19.4 m, 2/3 of which ranged from 7.5 to 12.9 m.

Therefore, the initial maximum lag distance for all experimental variograms was 10 m, and was applied as the "cutoff" parameter in the "variogram" function. The number of sample pixels for which to calculate semivariance at each lag was set to 100 ("size" = 100). The last parameter, lag size ("lag"), was determined by testing several values in an attempt to produce an experimental variogram with a sill and range, and low nugget. The first lag size tested was 0.66 m, which was determined by dividing the maximum lag distance by 15 (Geldsetzer, 2018b). Additional tested lag sizes were 0.30 m, 0.10 m, and the image resolution, 0.02706 m.

By visually inspecting the experimental variograms produced using each lag size, a suitable lag size was chosen. To produce the final variogram models, a variogram cloud was generated for each sample. For each variogram cloud, the maximum lag distance ("cutoff") was 1/3 to 2/3 of the maximum diagonal length of the sample building. For each cloud, the number of sample pixels for which to calculate semivariance at each lag was 1000 ("size"=1000), and the lag distance ("lag") was the value chosen from the initial experimental variograms. Each variogram cloud was exported from R to a comma separated values (csv) spreadsheet. In each spreadsheet, the mean semivariance was calculated for each lag – this resulted in a table representing the experimental variogram (Geldsetzer, 2018b). Each experimental variogram table was then imported into Matlab, where a power curve with 2 terms was fit to produce a variogram model. Matlab curve fitting, as well as other software options, did not offer spherical model fitting, so variogram models with ranges could not be produced. Instead, the power curve was used to estimate the range of each variogram model. With a range of variogram ranges from the 10 variogram models, a suitable texture window size was determined.

In addition to the power model, a Gaussian curve was also fitted to the experimental variograms. The goal with the Gaussian model was to identify initial flatness in the model. If initial

flatness existed, then the lag distance at which the flatness ended would signify a suitable maximum pixel size for texture analysis.

3.4 GLCM texture image generation

PCI Geomatics Geomatica 2017 was used to generate 8 GLCM texture images (each one using a different GLCM texture). The TEX (texture analysis) module was used (PCI Geomatics, 2017). As explained in the previous section, the variogram analysis found a suitable window size. The number of grey levels was set to 16, as suggested by Hall-Beyer (2017a; 2016b). The spatial relationship between the reference pixel and neighbor pixel was direction-invariant (omnidirectional), with a one-pixel offset (adjoining pixels) (Hall-Beyer, 2017b). A second set of texture images was produced with a larger window size and the same number of grey levels and spatial relationship, in order to observe the effects of a larger window on damage classification accuracy.

3.5 Final data spreadsheet

In ESRI ArcMap 10.5, the training and testing building footprints were used to extract zonal statistics from each texture image. From the zonal statistics, the mean texture value for each building was used to represent the building texture. This statistic was chosen because it describes the central tendency of the texture values within buildings, and not the less frequent values. For example, while some undamaged buildings contained areas of high contrast at edges of objects and paint strips (e.g., sample buildings #16, #17, #18 in Figure 5), their central tendency was low contrast. Similarly, while some damaged buildings contained areas of low disorder within large wood panels or other large materials (e.g., sample buildings #4, #5, and #6 in Figure 4), their central tendency was high disorder.

The mean texture values were joined to the building footprint shapefile. The shapefile attribute table was exported into training and testing comma separated values (csv) spreadsheets, with the following fields: building ID, roof damage (0 or 1), mean contrast, mean dissimilarity, mean homogeneity, mean angular second moment, mean entropy, mean GLCM mean, mean GLCM standard deviation, and mean GLCM correlation. Each mean texture field occurred twice, with one for each window size.

3.6 Descriptive statistics

Before building the logistic regression model, it was necessary to produce descriptive statistics to describe the independent variables, i.e., each mean texture measure. For the mean texture measures pertaining to the smaller window size, five-number summaries, ranges, means, standard deviations, histograms, normal Q-Q plots, and a correlation matrix were generated using R. The histograms and normal Q-Q plots were generated to identify potential outliers, and to determine whether each independent variable was normally distributed. The distribution of each independent variable would affect the choice of correlation statistic used in the correlation matrix (i.e., Pearson's or Spearman's).

3.7 Checking logistic regression assumptions

The assumptions of logistic regression are that: (i) the conditional probabilities are related to the independent variable(s) through a logistic function, (ii) all important independent variables are included in the model, (iii) no unimportant independent variables are included, (iv) there are minimal measurement errors, (v) the observations are independent, and residuals are not autocorrelated, (vi) multicollinearity is absent, and (vii) strong outliers are absent (Geldsetzer, 2018a).

The first assumption, (i), cannot be checked. Regarding the assumptions of the model containing all important and no unimportant independent variables (ii-iii), the latter is easier to adhere to than the former. Regressing GLCM texture measures against the presence or absence of roof damage is based on previous studies, so the inclusion of the mean texture measures as independent variables is logical. However, there are other factors that are logical predictors of roof damage, such as construction materials of the building, the wind speed that impacted the building, and perhaps the proximity of nearby buildings (to provide protection). The assumption of minimal measurement error, (iv) was checked by inspecting each building roof damage value (0 or 1) for accuracy. The calculation of texture values for each building can be checked, and PCI Geomatics does provide the equations it uses for each texture (PCI Geomatics, 2017). However, this would be time-consuming. To check for independence of errors (assumption v), a Global Moran's I can be calculated on the model residuals. To check for and resolve issues with multicollinearity (assumption vi), first R was used to perform 8 univariate logistic regressions, each using a different texture measure as the independent variable. Then, the correlation matrix was used to identify strong correlation between two independent variables, which was defined as an absolute correlation of at least 0.70. From each group of two or more correlated variables, the one with the lowest significant p-value from the univariate logistic regression was kept. Thus, no correlated independent variables remained as candidate predictors for model building. Finally, to check for outliers (assumption vii), the normal Q-Q plot of each independent variable was inspected.

3.8 Logistic regression

Logistic regression models were generated in R using the training data (40 "undamaged" and 40 "damaged" building samples) and the function "glm". The family of logistic regression model was binomial, with a logit link function. To describe each model, summary and goodness-of-fit statistics were

produced, including predictor p-values, Akaike Information Criterion (AIC), McFadden R^2, and chi-square analysis of deviance. To assess the predictive ability of each model, the model was used to calculate the conditional probability of damage for each building in the testing dataset (40 "damaged" and 40 "undamaged" building samples). This was done using the "predict" function in R. Buildings with conditional probabilities less than or equal to 0.50 were classified as "undamaged", and buildings with values greater than 0.50 were classified as "damaged". A confusion matrix was constructed to show numbers of true positives, true negatives, false positives, and false negatives. The classification accuracy of each model was calculated by dividing the number of true positives and true negatives by the testing sample size, 80. A Receiver Operating Characteristic (ROC) curve was generated by plotting the false positive rate versus the true positive rate. Then, the area under the curve (AUC) statistic was calculated.

3.9 Random Forest classification

To compare against the logistic regression models, Random Forest (RF) models were generated in R using the package "randomForest". The RF type was classification (as opposed to regression). Each RF model was used to predict "damaged" or "undamaged" for each building in the testing dataset. A confusion matrix was generated, followed by a calculation of the classification accuracy.

4. Results

4.1 Variogram analysis

The experimental variograms constructed with various lag sizes are shown in Figure 6. For each sample building, the 0.66 m lag size produced an experimental variogram with the first lag containing a semivariance of at least approximately 1000 (Figure 6a). If models were fitted to these variograms, the nuggets would be large (Figure 6a). The 0.30 m lag size produced experimental variograms with lower

initial semivariances, albeit starting at values of approximately 600 (Figure 6b). The 0.10 m lag size produced experimental variograms with lower initial semivariances, starting at approximately 400 (Figure 6c). Finally, the 0.02706 m lag size produced experimental variograms with the lowest initial semivariances, starting at approximately 50 (Figure 6d).

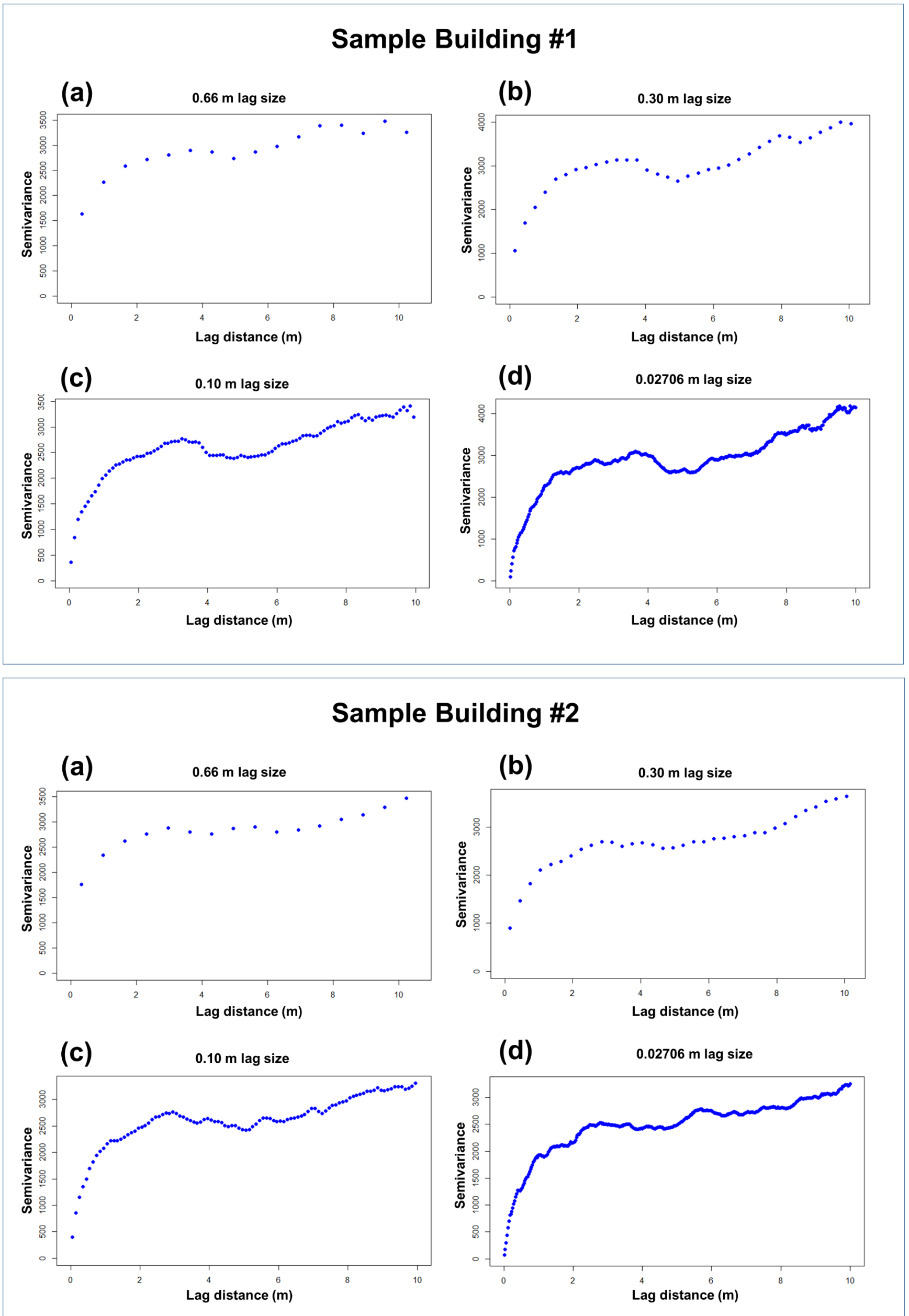
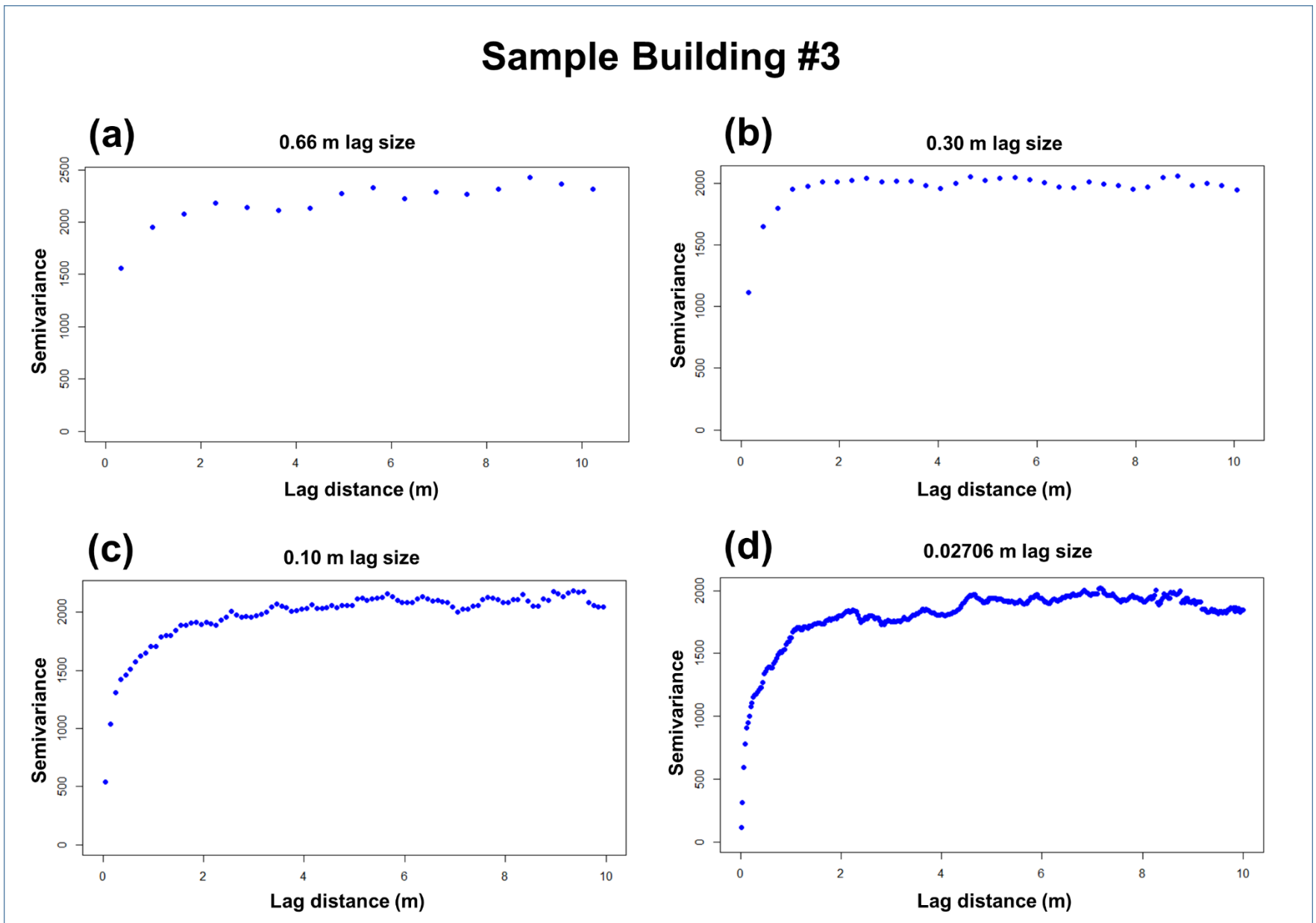


Figure 6. Experimental variograms generated for each sample damaged building using 100 samples and four different lag sizes: (a) 0.66 m, (b) 0.30 m, (c) 0.10 m, and the UAV orthomosaic resolution, 0.02076 m.



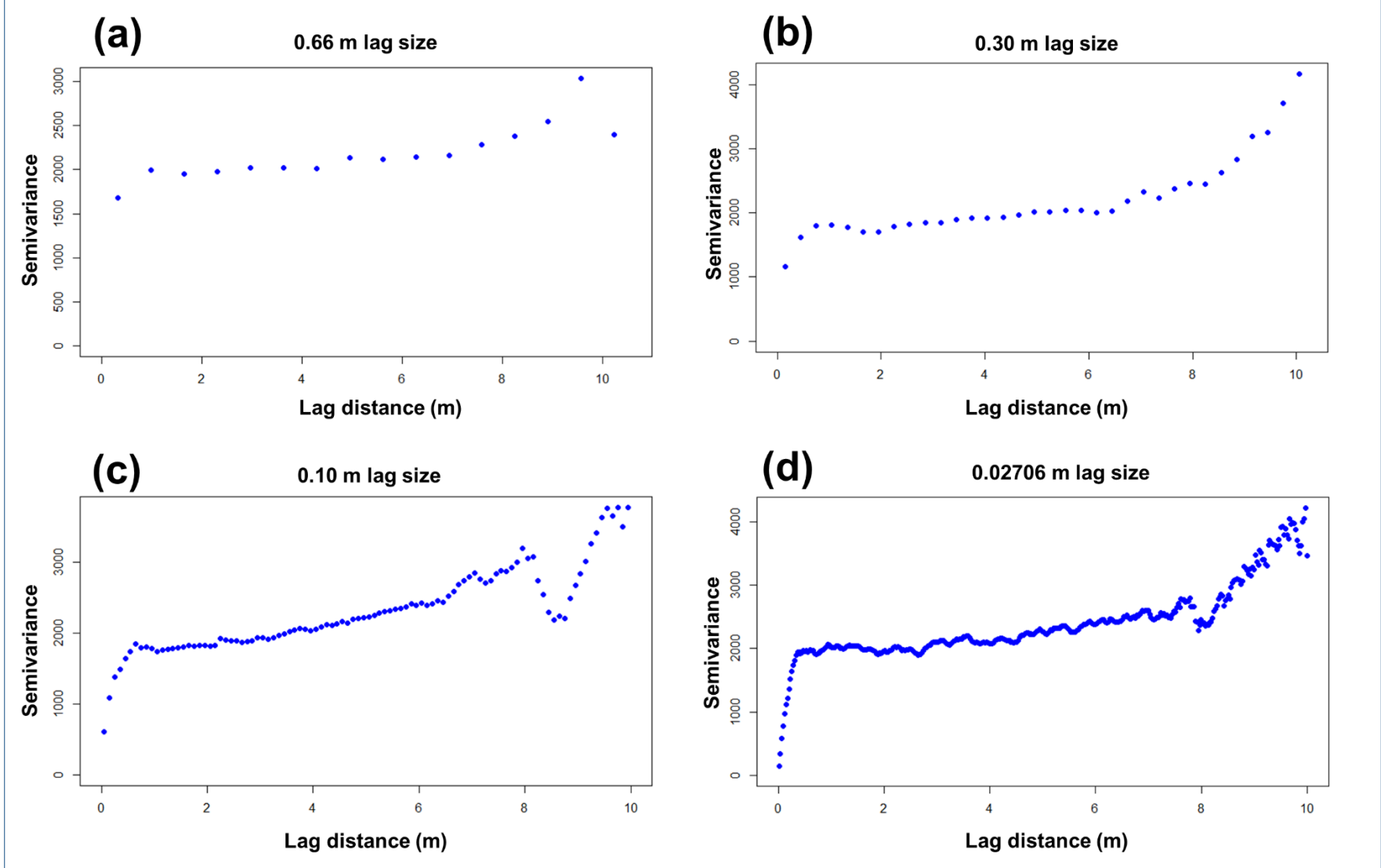
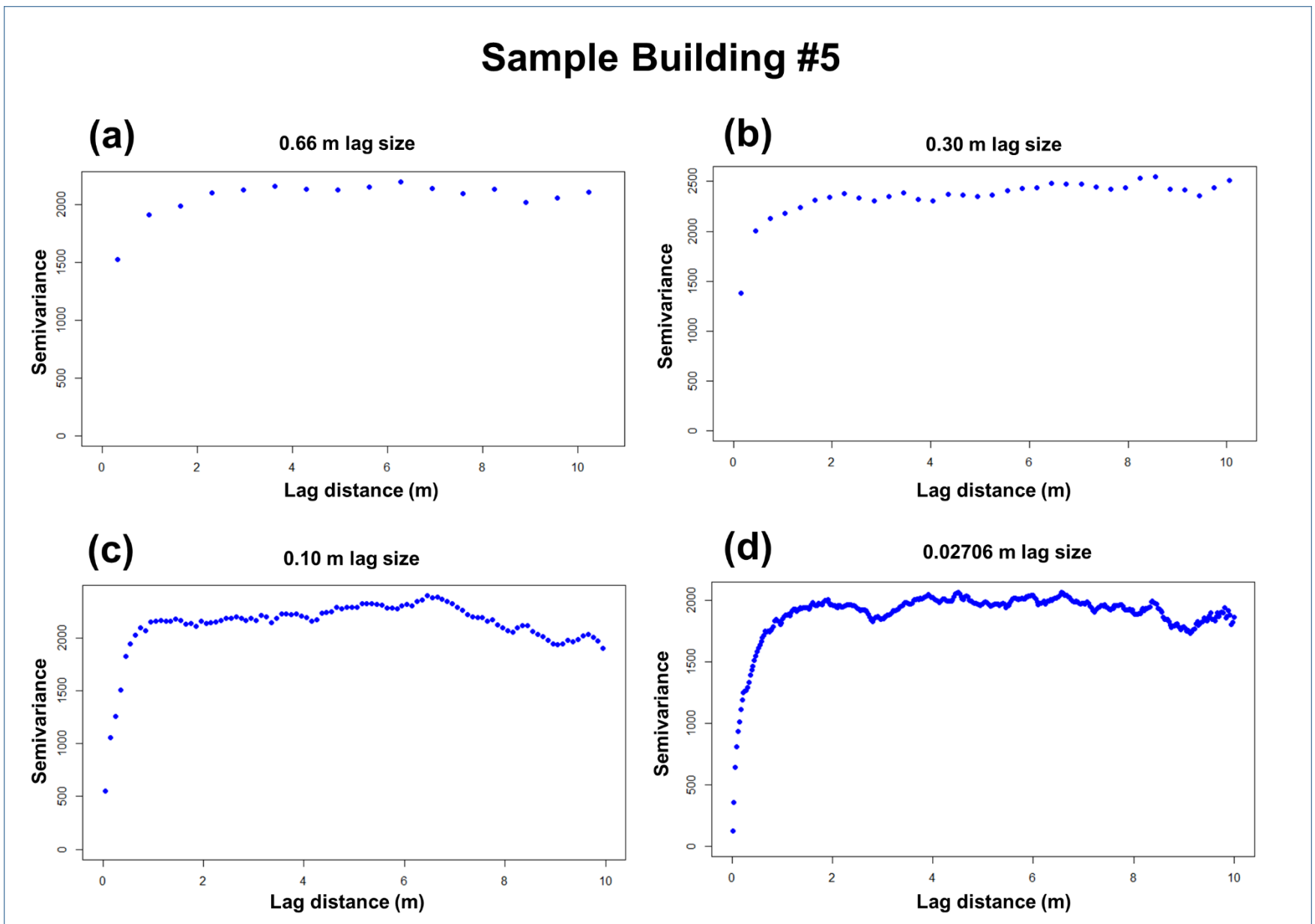


Figure 6 (continued)



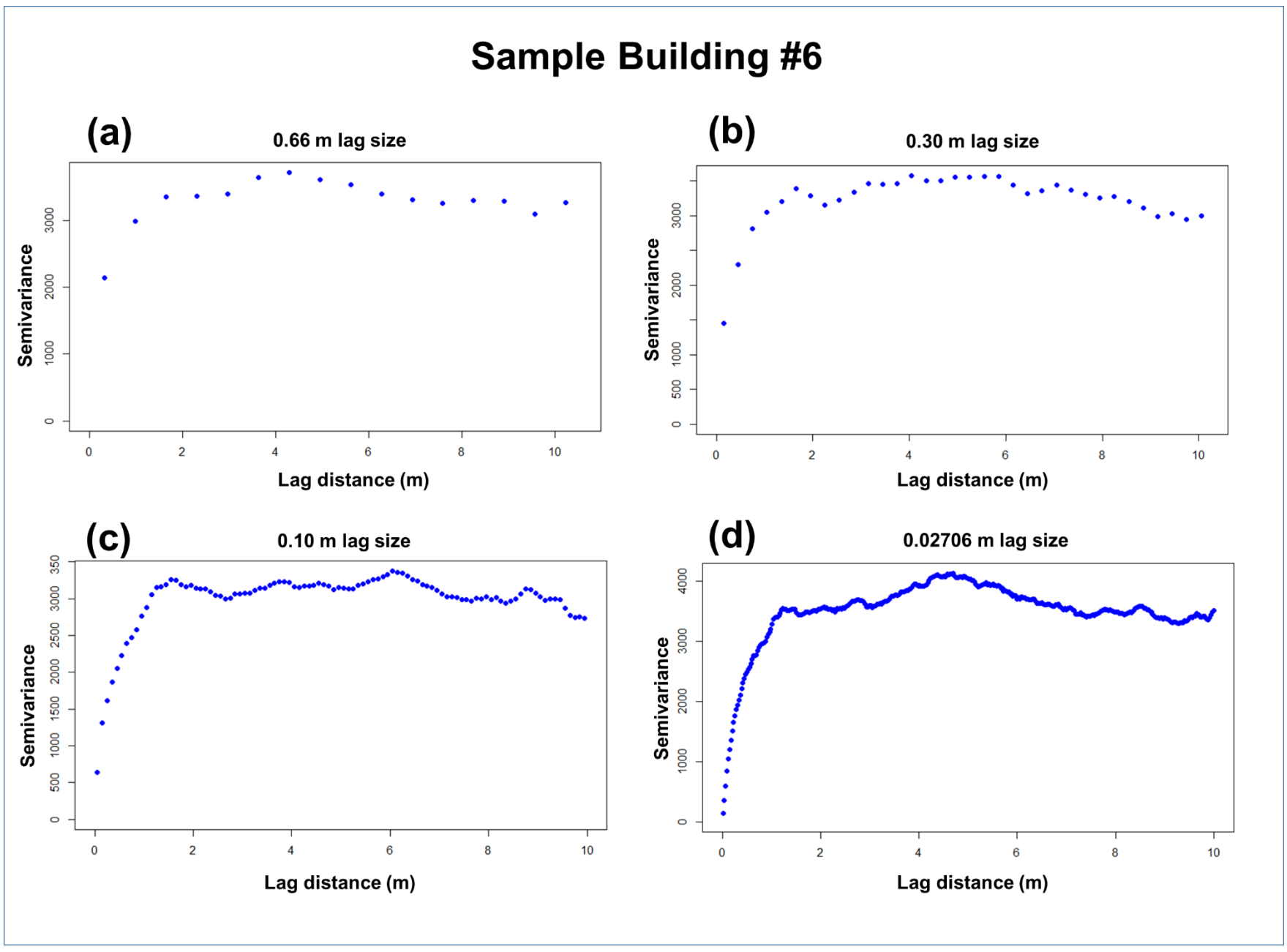


Figure 6 (continued)

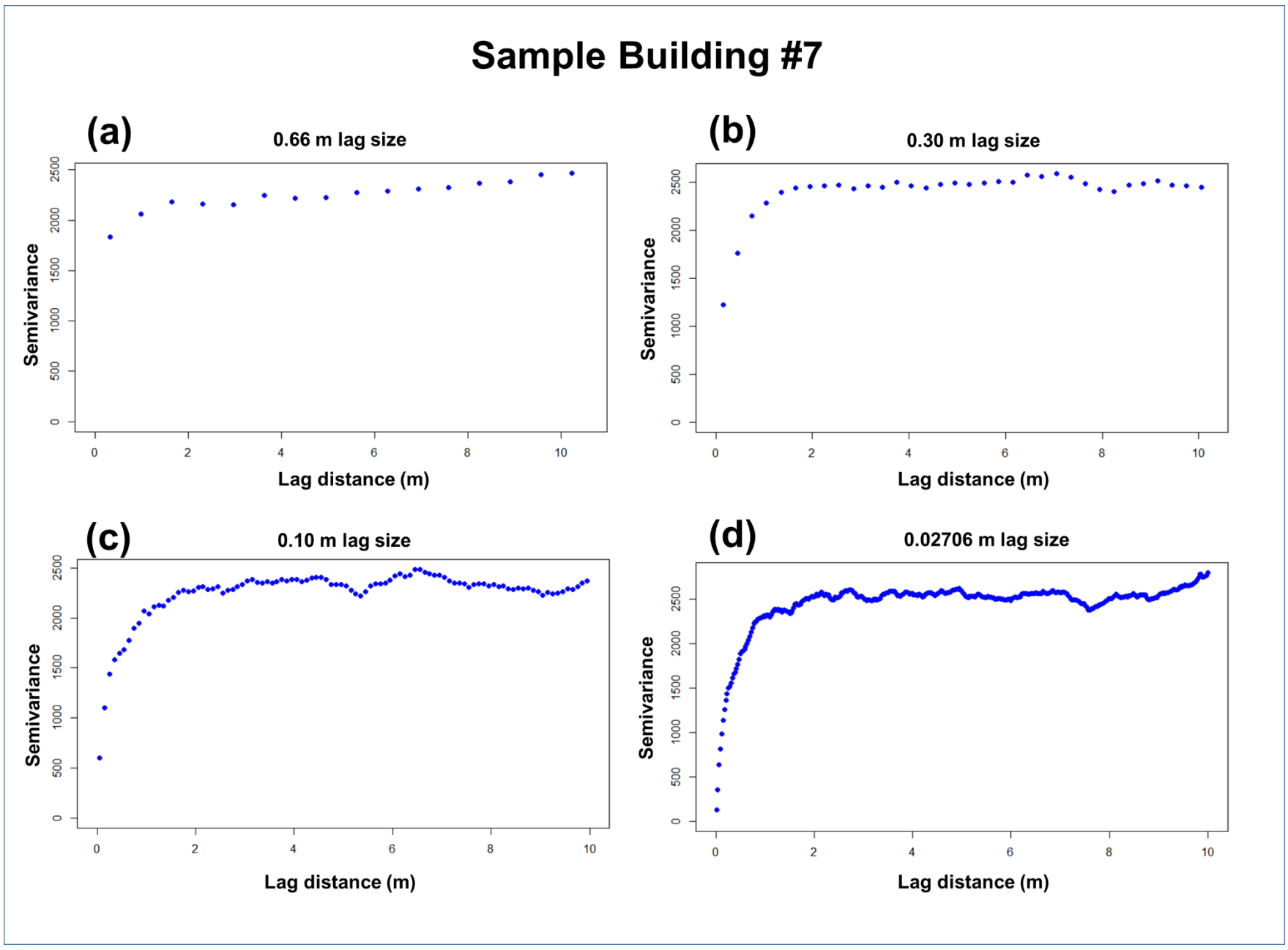
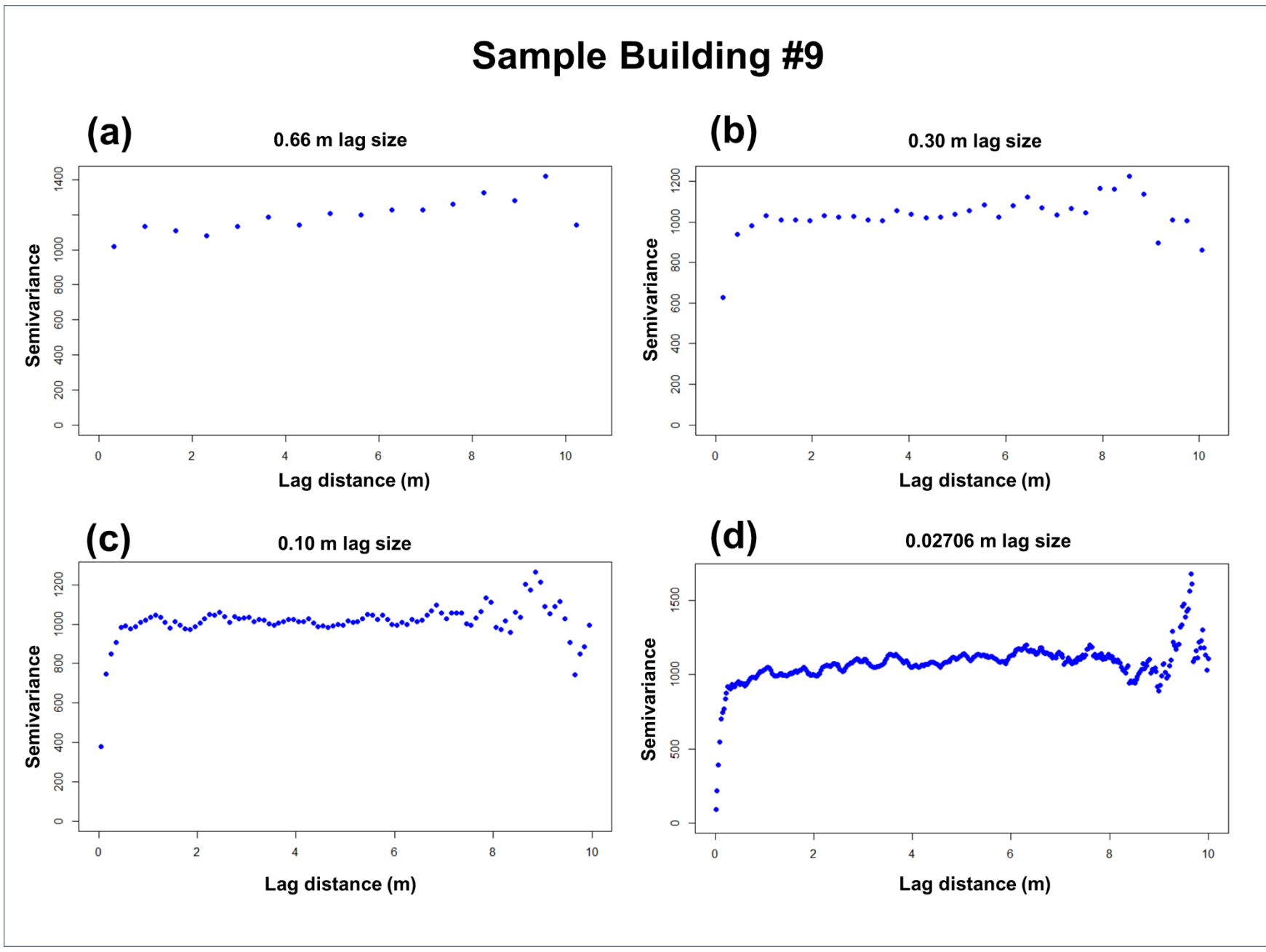


Figure 6 (continued)



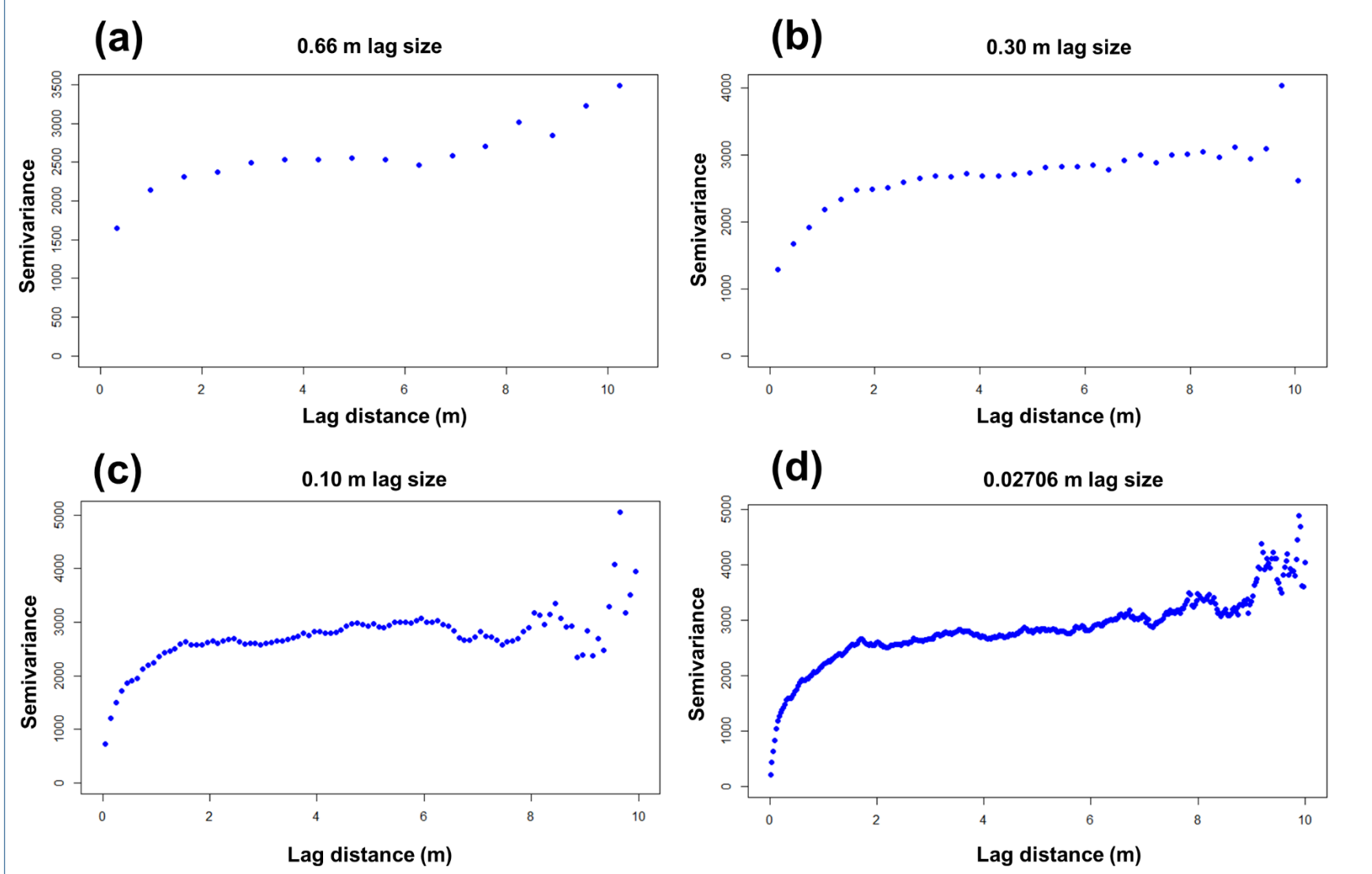
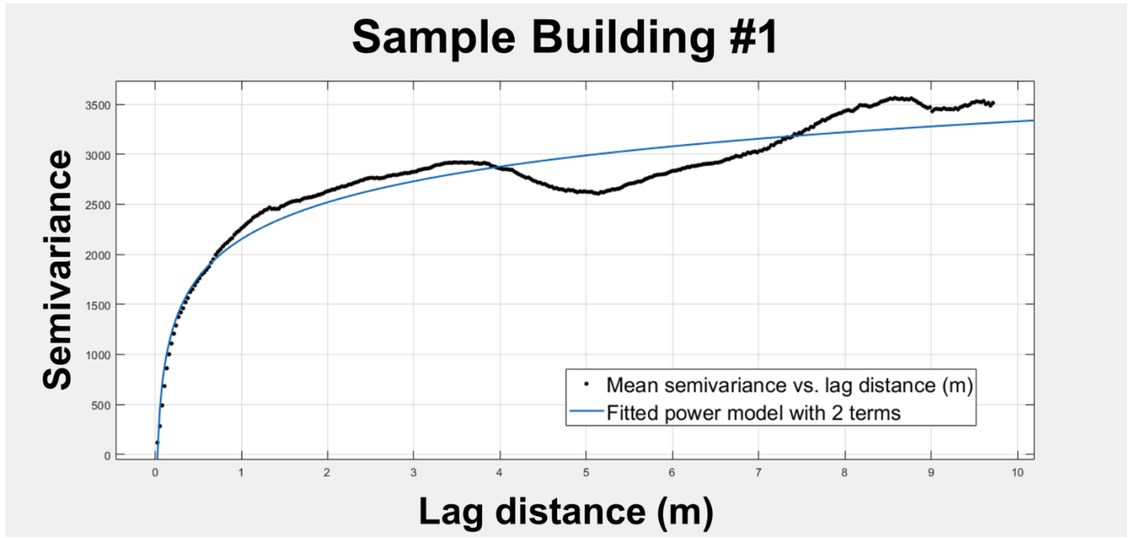


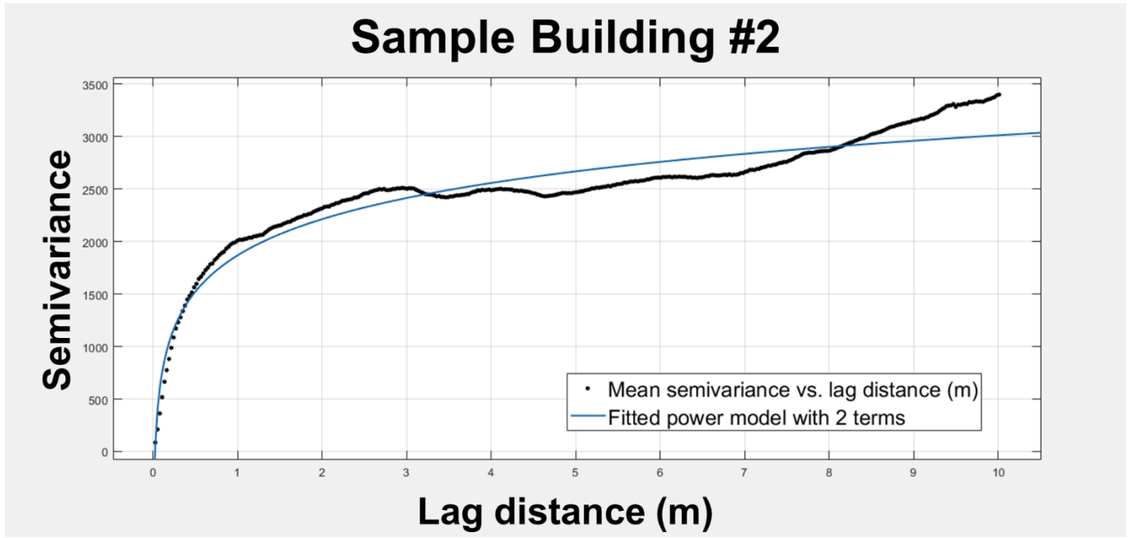
Figure 6 (continued)

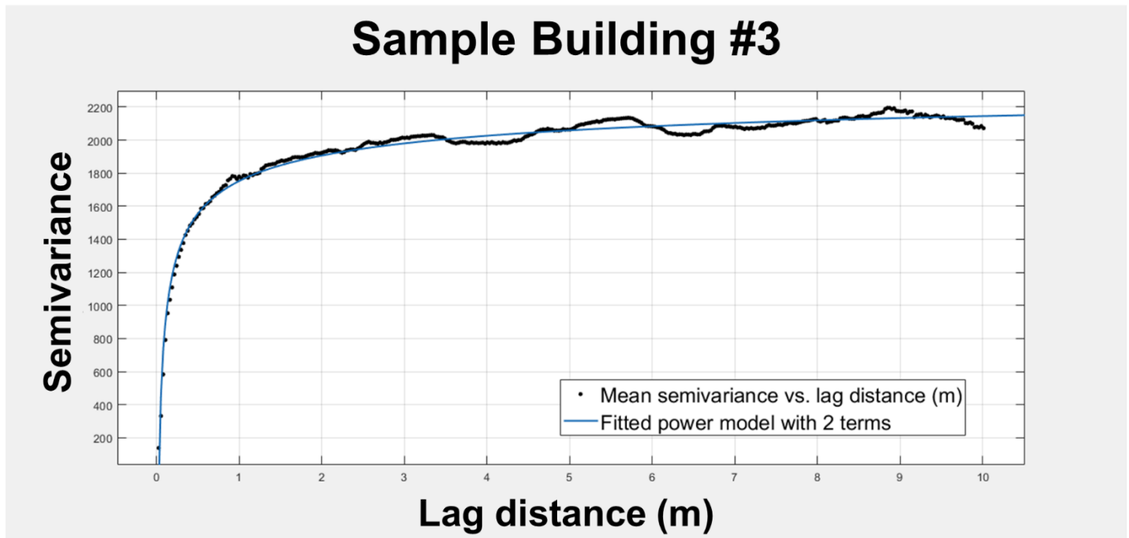
Because of the high nuggets the 0.66 m, 0.30 m, and 0.10 m lag sizes would produce if models were fitted to the experimental variograms, the 0.02706 m lag size was used to construct the final variogram models. As well, for the final variogram models, the maximum lag distance of samples #1, #4, #9, and #10 was decreased to 2/3 of their maximum diagonal lengths. The experimental variograms of these samples show spikes and erratic values of semivariance past 2/3 of their maximum diagonal lengths (Figure 6).

Figure 7 shows the final variogram models for each sample. For each variogram model, 1000 samples were used to compute semivariance at each lag, 0.02706 m was the lag size, the maximum lag distance was 1/3 to 2/3 the maximum diagonal length of the building, and a power curve with 2 terms was fit (Figure 7). Due to the nature of the power model, the semivariance steadily increases until the maximum lag in all the variogram models, although at a different rate for each sample (Figure 7). As well, most, if not all, the variogram models show a negative nugget, so the 0.10 m lag size likely would have been more appropriate (Figure 7). The ranges of the variogram models had to be approximated since spherical models could not be fit. The ranges were approximated by identifying the lag distance at which the slope of the curve began to drop below 45°. By this definition, the variogram ranges ranged from approximately 0.5 to 1.0 m (Figure 7). With a lag size of 0.02706 m, these variogram ranges ranged from to 19 to 37 pixels. Therefore, the window size for texture calculation was greater than the maximum variogram range, at 51 x 51 pixels (1.4 x 1.4 m). A larger window size of 101 x 101 pixels (2.7 x 2.7 m) was also tested to compare damage classification accuracy.

For the second objective of the variogram analysis, Gaussian models were fit to the variograms. However, none of the Gaussian models exhibited initial flatness. This is apparent in all the variograms, where the semivariance values at the initial lags form steep slopes (Figure 7). Therefore, it was not possible to identify a maximum suitable pixel size for this application.







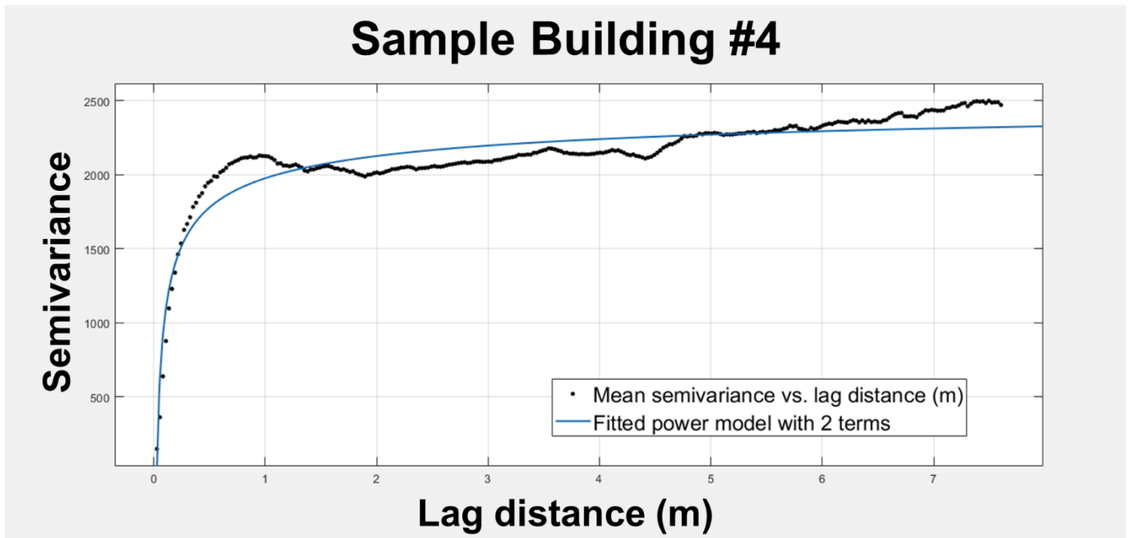
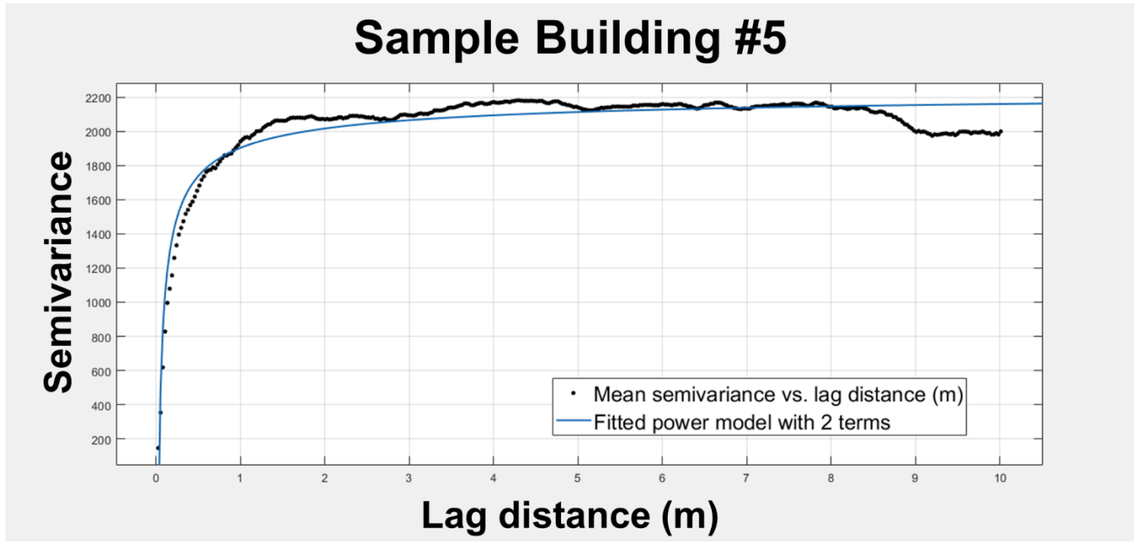
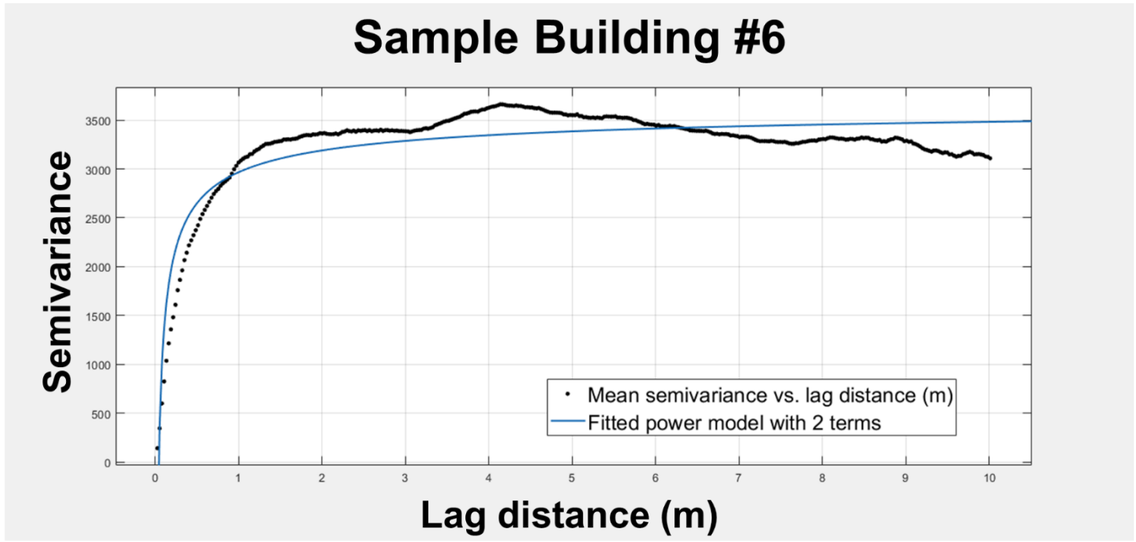
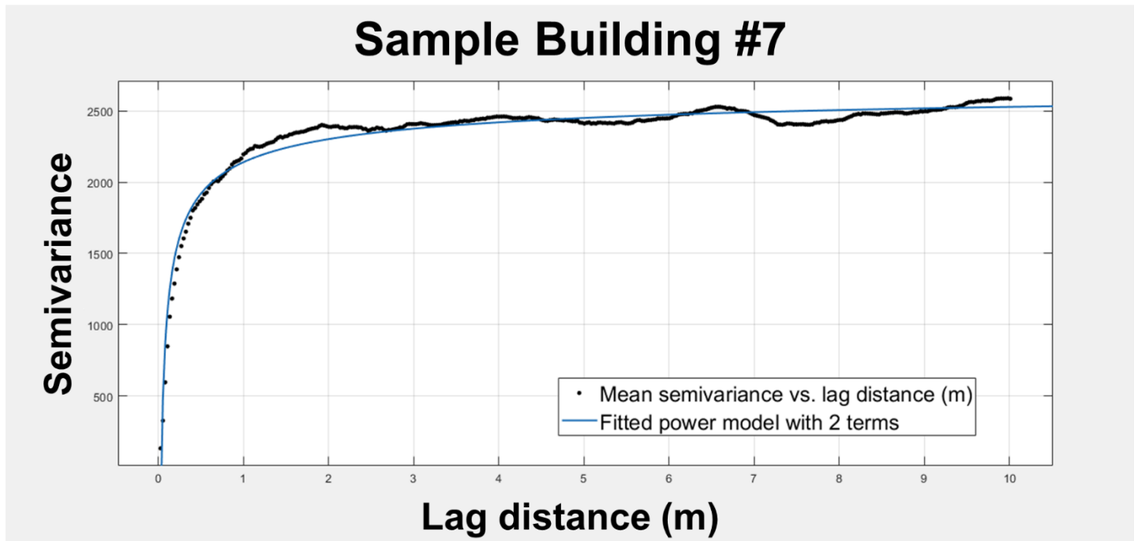


Figure 7. Empirical variogram model for each sample damaged building.







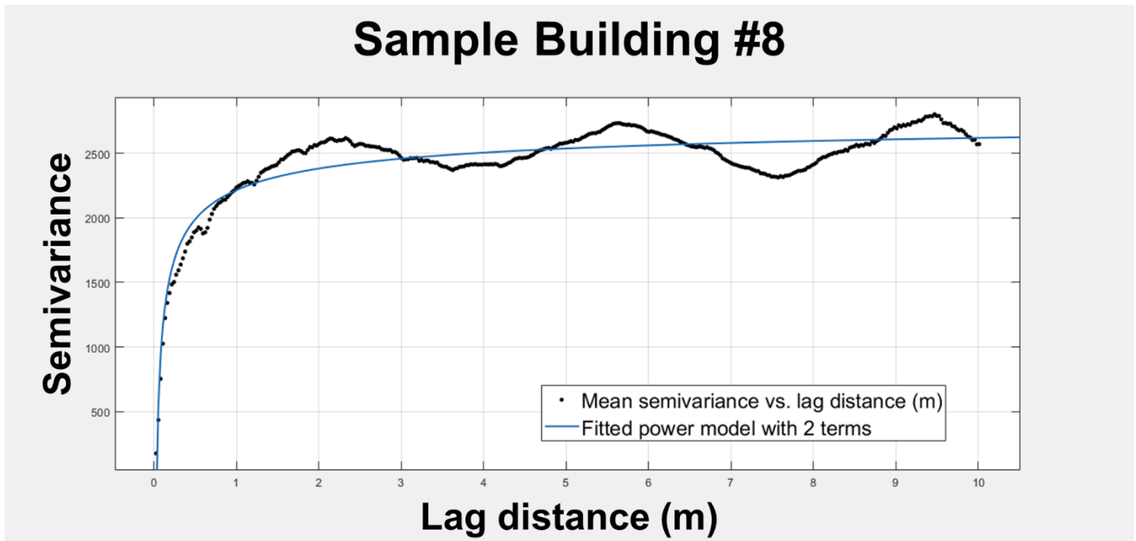
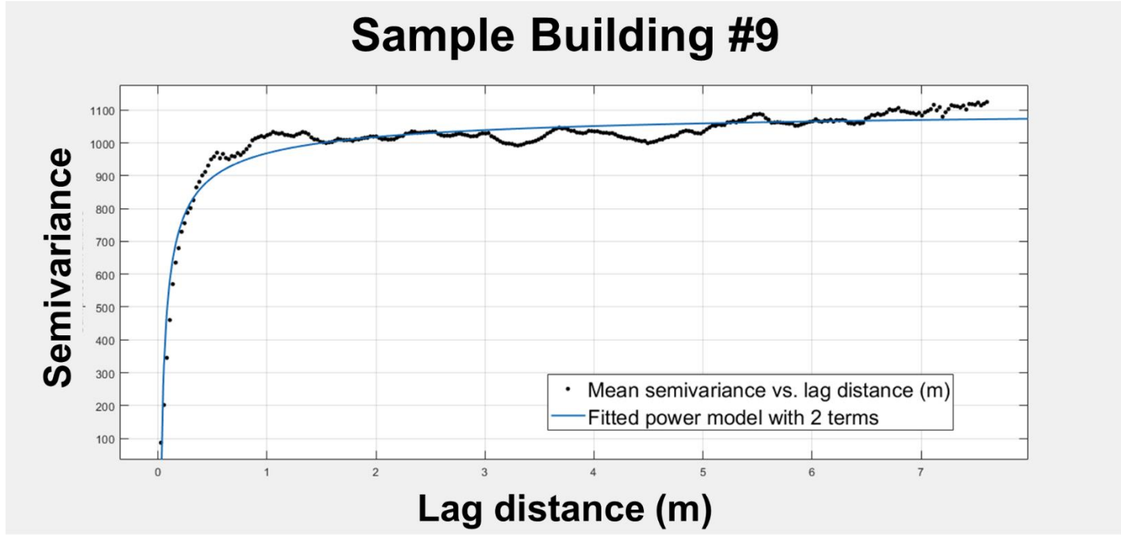


Figure 7 (continued)



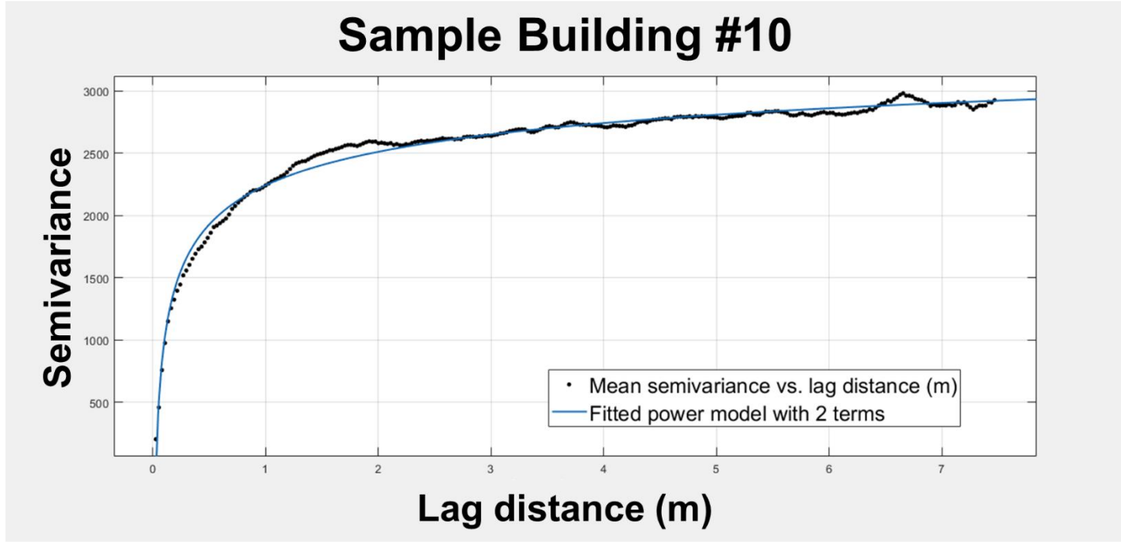


Figure 7 (continued)

4.2 Descriptive statistics

Eight GLCM texture images were calculated using a 51 x 51 pixel window size, as well as 8 GLCM texture images using a 101 x 101 pixel window size. For each image, the mean texture value within each training and testing sample building footprint was calculated. Descriptive statistics were produced for the mean texture values (51 x 51 pixel window) within the training building footprints. The five-number summaries, ranges, means, and standard deviations of the independent variables are shown in Table 1. The histograms and normal Q-Q plots are shown in Figure 8. The histograms show that the distributions are either bimodal or skewed (Figure 8). Bimodal distributions are logical; for example, damaged buildings may tend to have lower homogeneity, while undamaged buildings may tend to have higher homogeneity (Figure 8). Mean contrast and mean GLCM correlation contain potential outliers, as shown by the normal Q-Q plots (Figure 8). Table 2 shows the correlation matrix; since the independent variables are non-normally distributed, only the Spearman's correlation coefficient was considered (Rogerson, 2006). A minimum Spearman's value of 0.70 (70%) was used to identify potential multicollinearity (Geldsetzer, 2018c), as shown by the red highlighting in Table 2. Mean GLCM mean and mean GLCM correlation were not correlated with other independent variables, while the others were all correlated with one another.

Table 1. Descriptive statistics of each independent variable (training samples, 51x51 pixel window).

Independent variable	Min	Max	Range	Q1	Median	Q3	Mean	Std Dev
Mean contrast	0.26	7.75	7.49	0.75	1.77	3.35	2.09	1.54
Mean dissimilarity	0.16	1.76	1.61	0.37	0.78	1.13	0.77	0.41
Mean homogeneity	0.52	0.94	0.42	0.60	0.69	0.84	0.72	0.12
Mean ASM	0.02	0.70	0.68	0.04	0.08	0.26	0.16	0.17
Mean entropy	0.89	4.48	3.58	2.13	3.33	4.11	3.12	1.08
Mean GLCM mean	1.44	14.35	12.91	5.23	6.44	8.01	7.05	2.59
Mean GLCM std. dev.	0.66	4.39	3.73	1.55	2.70	3.65	2.63	1.12
Mean GLCM correlation	0.46	0.92	0.46	0.79	0.85	0.88	0.82	0.09

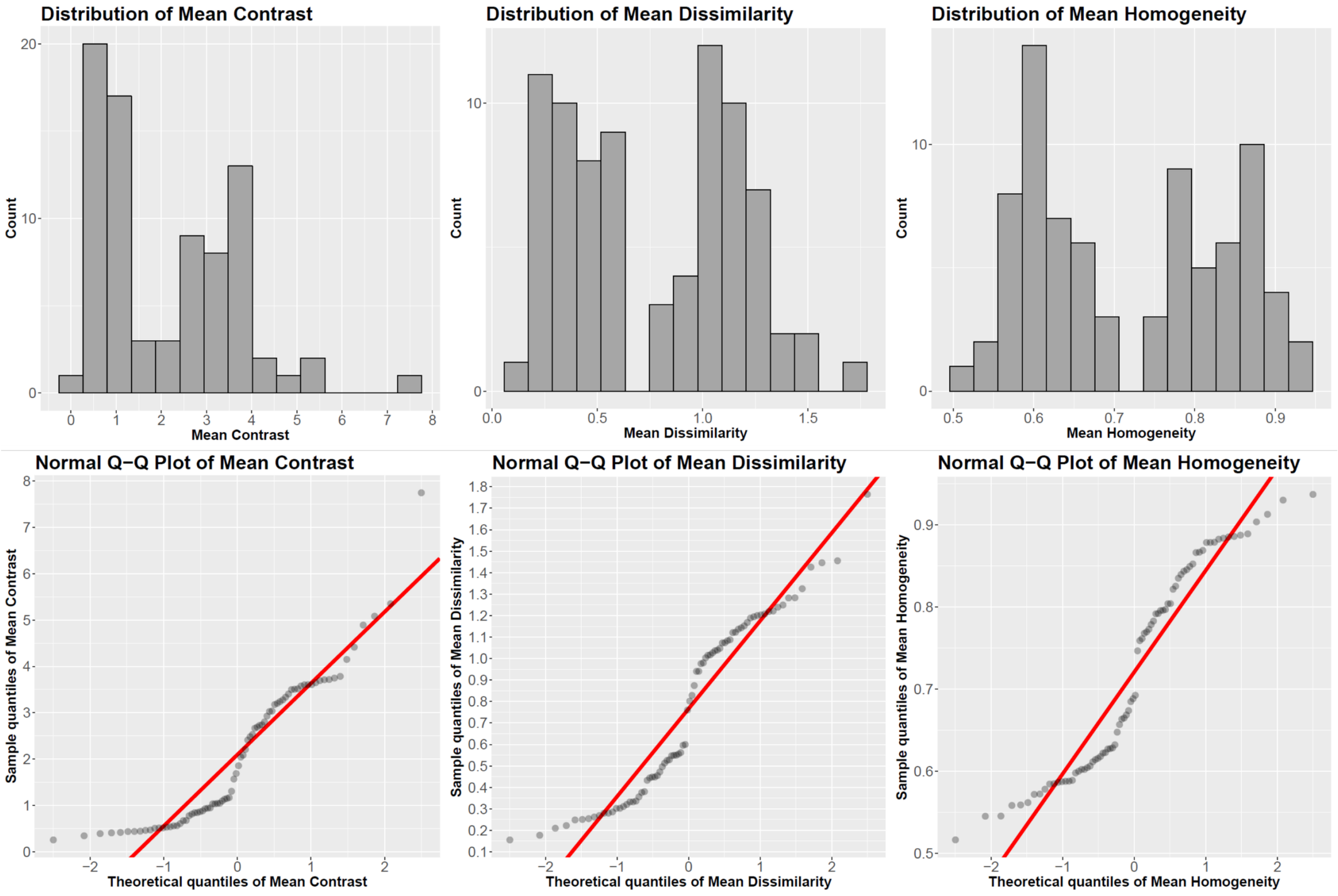


Figure 8. Histograms and normal Q-Q plots of each independent variable (training samples, 51x51 pixel window).

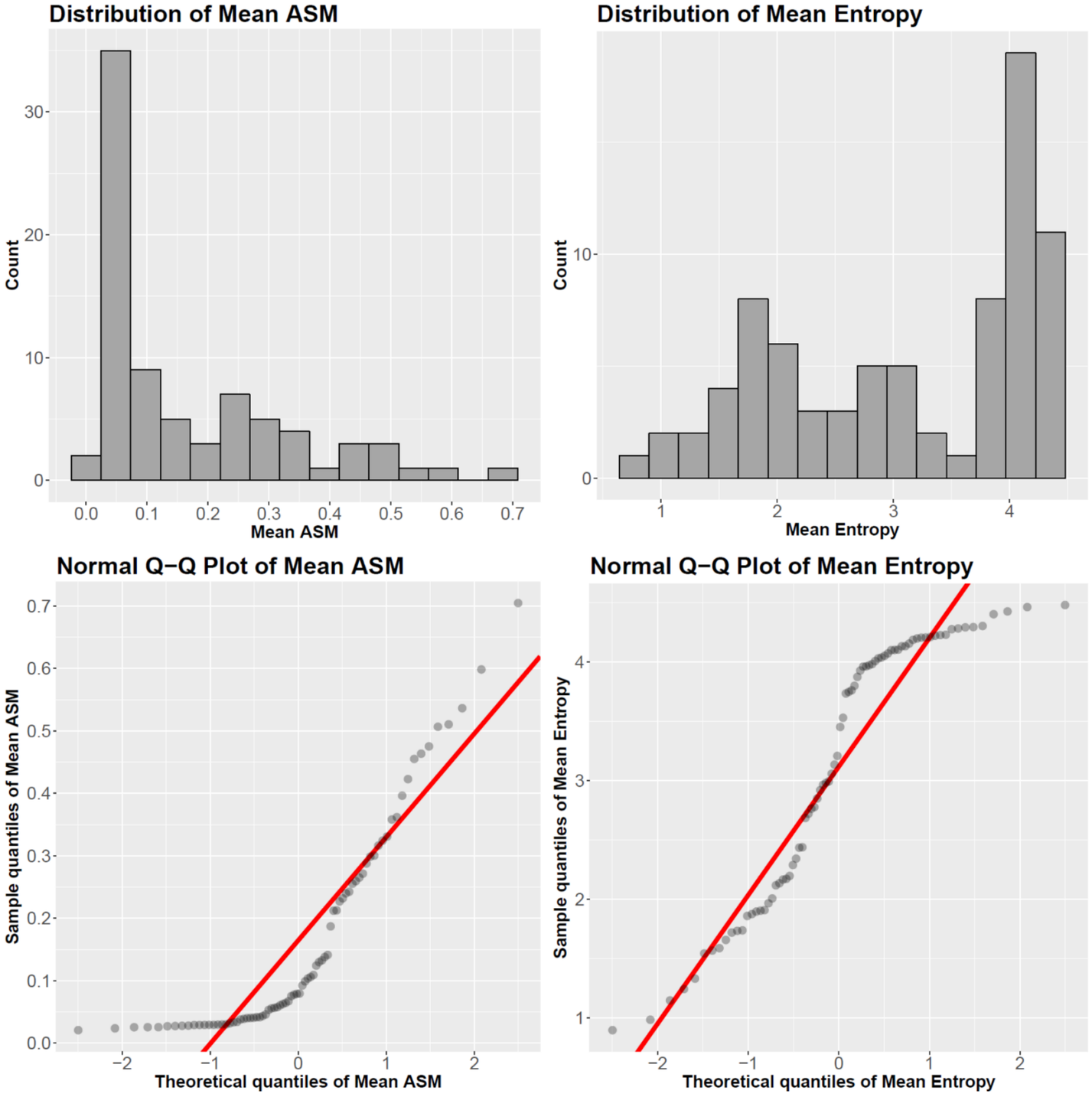


Figure 8 (continued)

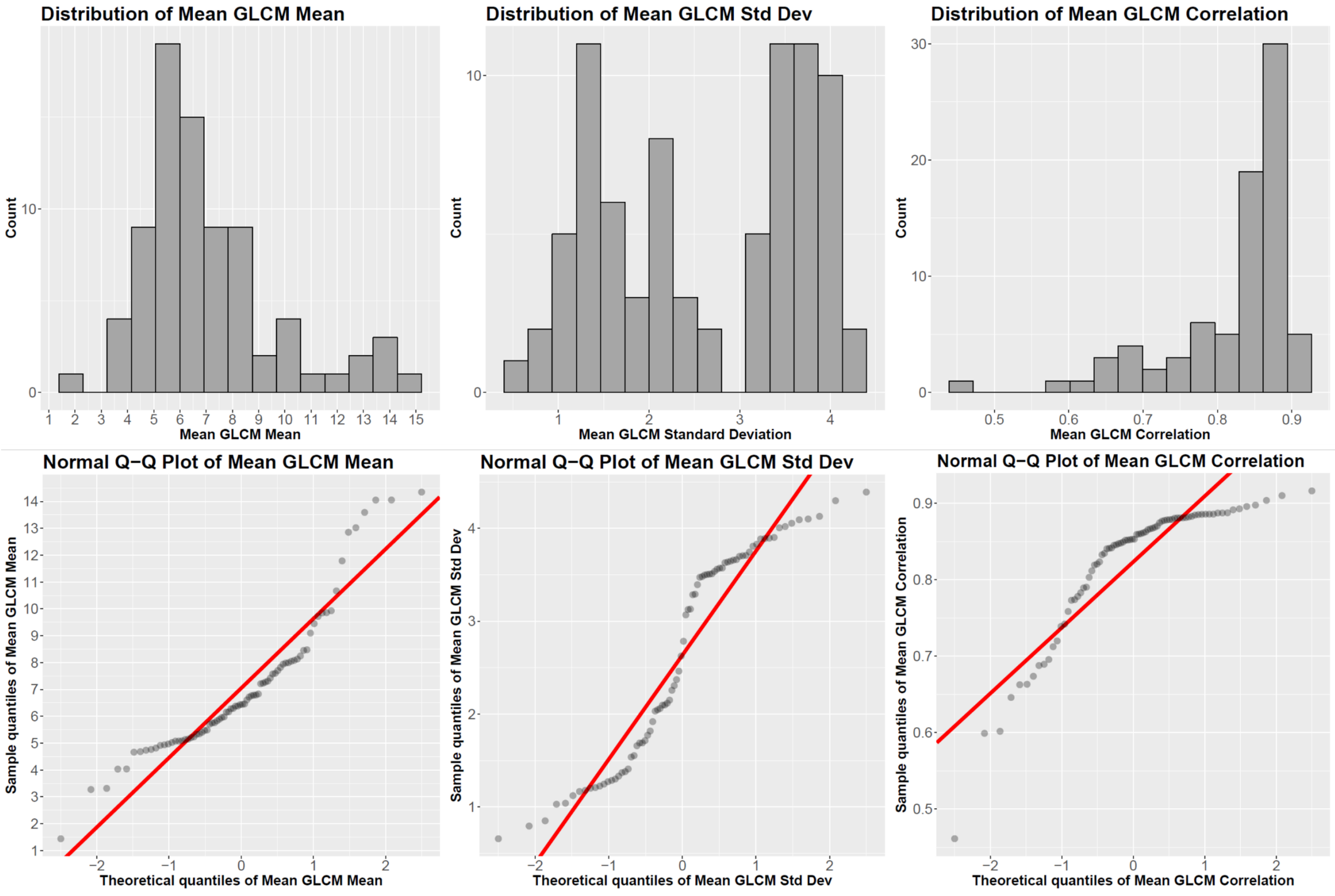


Figure 8 (continued)

Table 2. Correlation matrix containing Spearman's correlation values for each pair of independent variables. High correlation values (≥0.70) are highlighted in red.

	Mean contrast	Mean dissimilarity	Mean homogeneity	Mean ASM	Mean entropy	Mean GLCM mean	Mean GLCM std. dev.	Mean GLCM correlation
Mean contrast	1	0.98	-0.95	-0.85	0.93	-0.36	0.94	0.47
Mean dissimilarity	0.98	1	-0.99	-0.91	0.97	-0.39	0.91	0.47
Mean homogeneity	-0.95	-0.99	1	0.95	-0.99	0.41	-0.87	-0.47
Mean ASM	-0.85	-0.91	0.95	1	-0.97	0.40	-0.80	-0.56
Mean entropy	0.93	0.97	-0.99	-0.97	1	-0.38	0.89	0.54
Mean GLCM mean	-0.36	-0.39	0.41	0.40	-0.38	1	-0.31	-0.12
Mean GLCM std. dev.	0.94	0.91	-0.87	-0.80	0.89	-0.31	1	0.64
Mean GLCM correlation	0.47	0.47	-0.47	-0.56	0.54	-0.12	0.64	1

4.3 Logistic regression

Univariate logistic regressions were performed by regressing each independent variable with the binary dependent variable (i.e., the 0 and 1 values pertaining to undamaged and damaged buildings, respectively, of the training dataset). The results of the univariate logistic regressions, models #1 through #8, are shown in Table 3. The significant ($p<0.05$) independent variables were mean contrast, mean ASM, mean GLCM mean, mean GLCM standard deviation, and mean GLCM correlation (Table 3). Because mean GLCM mean and mean GLCM correlation were not highly correlated with other independent variables (Table 2), and because they were both significant predictors of damage, they were added to a multivariate logistic regression model (#9) (Table 3). From the remaining significant candidate predictors, the variable with the lowest p-value, mean GLCM standard deviation (Table 3), was added as the third and final predictor of model #9 (Table 3).

A backward stepwise approach was employed for the final logistic regression model building, where all three predictors were initially included in the model. In this model (#9), the predictor coefficients had high standard errors, and the predictors were non-significant ($p=1.00$); therefore, model #10 was created by eliminating the predictor with the highest p-value in the univariate logistic regression, mean GLCM standard deviation (Table 3). Model #10 was the final model since both predictors were significant (Table 3). In addition to the predictor coefficients and p-values, the chi-square p-value was calculated to test the significance of the model's residual decrease caused by the addition of each predictor (Table 3). For model #10, the null deviance (i.e., of the model with only an intercept) was 110.90, and was decreased to 86.23 with the addition of the mean GLCM mean predictor, and was further decreased to 46.12 with the addition of the mean GLCM correlation predictor (Table 3). Both predictor residual deviances were significant ($p<0.05$) (Table 3).

Table 3. Univariate and multivariate models with logistic regression summary statistics, goodness-of-fit statistics, and classification accuracies, all pertaining to the texture predictors derived from a 51 x 51 pixel window. Classification accuracies of logistic regression models whose texture predictors were derived from a 101 x 101 pixel window are also shown. For comparison to the logistic regression models, Random Forest classification accuracies (for the 51 x 51 pixel window predictors) are also shown.

		51 x 51 pixel window for texture calculation													
Model #	**Model predictor(s)**	**Slope coefficient**	**Std error**	**z-value**	**p-value**	**Null deviance (79 d.f.)**	**Residual deviance (78 d.f.)**	**Chi sq. p-value**	**R warning 'fitted probabilities numerically 0 or 1 occurred'**	**McFadden R^2**	**AIC**	**AUC**	**Classification accuracy (%)**	**101 x 101 pixel window classification accuracy (%)**	**Random Forest classification accuracy (51 x 51 pixel window) (%)**
1	**Mean contrast**	7.04	3.05	2.31	0.021	110.90	7.13	< 2.2e-16	Yes	0.94	11.13	0.72	67.5	67.5	90.0
2	**Mean dissimilarity**	27.27	16.05	1.70	0.0892	110.90	4.72	< 2.2e-16	No	0.96	8.72	0.73	67.5	67.5	92.5
3	**Mean homogeneity**	-109.05	89.11	-1.22	0.221	110.90	5.15	< 2.2e-16	No	0.95	9.15	0.74	68.8	68.8	92.5
4	**Mean ASM**	-154.55	65.83	-2.35	0.0189	110.90	9.57	< 2.2e-16	Yes	0.91	13.57	0.74	71.3	71.3	92.5
5	**Mean entropy**	192.80	457184.00	0.00	1	110.90	0.00	< 2.2e-16	Yes	1.00	4.00	0.69	70.0	70.0	97.5
6	**Mean GLCM mean**	-0.63	0.17	-3.67	0.00024	110.90	86.23	0.00	No	0.22	90.23	0.58	53.8	53.8	71.3
7	**Mean GLCM std dev**	5.34	1.49	3.60	0.000324	110.90	13.89	< 2.2e-16	No	0.87	17.89	0.69	67.5	63.8	90.0
8	**Mean GLCM correlation**	28.29	7.72	3.67	0.000246	110.90	77.94	0.00	No	0.30	81.94	0.56	47.5	45.0	60.0
	Mean GLCM mean	mean: -22.35	mean: 58374.40	mean: 0	mean: 1		mean: 86.231	mean: 0.00							
	Mean GLCM correlation	cor: 237.78	cor: 1776504.13	cor: 0	cor: 1		cor: 46.119	cor: 0.00							
9	**Mean GLCM std dev**	std: 60.94	std: 149743.06	std: 0	std: 1	110.90	std: 0.000	std: 0.00	Yes	1.00	8.00	0.68	70.0	66.30	93.8
	Mean GLCM mean	mean: -1.5885	mean: 0.4521	mean: -3.514	mean: 0.00		mean: 86.231	mean: 0.00							
10	**Mean GLCM correlation**	cor: 47.7955	cor: 14.1245	cor: 3.384	cor: 0.00	110.90	cor: 46.119	cor: 0.00	Yes	0.58	52.12	0.61	63.8	51.30	70.0

The McFadden R^2 was also calculated for each model (Table 3). This value can be interpreted as the ratio of the null model's residual deviance that is accounted for by the inclusion of the predictor(s). For example, in model #10, the predictors decreased the null model's residual deviance by 64.78, which is 58% of the null model's residual deviance, resulting in a McFadden R^2 of 0.58 (Table 3). The AIC was also calculated for each model (Table 3). It is important to note that model #10 does not have a higher McFadden R^2 or lower AIC than a majority of the univariate models (Table 3).

The AUC and classification accuracy was calculated for each model by calculating conditional probabilities of damage for a testing dataset composed of 40 damaged buildings and 40 undamaged buildings (Table 3). Again, model #10 does not have a higher AUC or classification accuracy than a majority of the univariate models (Table 3).

Models #1 through #10 were generated again, this time using predictor variables that were calculated from a 101 x 101 pixel window for the texture images. Table 3 shows the classification accuracies of the models resulting from the larger texture window. The classification accuracies did not change when increasing the window size for models #1 through #6, but decreased with the larger window size for models #7 through #10 (Table 3).

4.4 Random Forest classification

Table 3 shows the classification accuracies of the Random Forest models, which used the predictors derived from 51 x 51 pixel windows for texture. The Random Forest model classification accuracies were calculated using the same testing dataset that was used for the logistic regression models. Whereas the logistic regression model classification accuracies ranged from 47.5 to 71.3%, the Random Forest classification accuracies ranged from 60.0 to 97.5% (Table 3). For both logistic regression and Random Forest classification, the model with the lowest accuracy was model #8 (univariate, mean

GLCM correlation) (Table 3). The logistic regression model with the highest accuracy was model #4 (univariate, mean ASM), and the Random Forest model with the highest accuracy was model #5 (univariate, mean entropy) (Table 3).

5. Discussion

5.1 GLCM texture measures as predictors of roof damage

The hypothesis of this study was that GLCM textures pertaining to contrast (contrast, dissimilarity, homogeneity) and orderliness (ASM, entropy) would be strong predictors of roof damage. The univariate models using the contrast group textures as predictors of damage achieved classification accuracies ranging from 67.5 to 68.8%, and the univariate models using orderliness group textures achieved accuracies of 70.0 to 71.3% (Table 3). The contrast textures describe the differences in neighboring pixel values within a window, while the orderliness textures describe the degree of organization of pixel values in a window (Hall-Beyer, 2017a). It is logical that, in a 51 x 51 pixel (1.4 x 1.4 m) window, there was higher contrast and disorder of pixel values for the damaged buildings than the undamaged buildings.

It is important to note that, for the univariate models, dissimilarity, homogeneity, and entropy were non-significant predictors ($p>0.05$) (Table 3). Further research will be done to understand why some models received a warning in R that, 'fitted probabilities 0 or 1 occurred' (Table 3). Also, it is unclear why the univariate model using entropy (model #5) had a very high standard error, p-value of 1.00, yet still achieved one of the highest logistic regression classification accuracies (Table 3). Overall, the best logistic regression model was model #4, which used ASM as the predictor (significant, $p<0.05$), and achieved a McFadden R^2 of 0.91, AUC of 0.74, and the highest classification accuracy, 71.3% (Table 3).

Two of three GLCM textures from the descriptive statistics group (GLCM mean, GLCM correlation) performed more poorly as predictors of damage than the contrast and orderliness groups. The univariate models using GLCM mean and GLCM correlation had low McFadden R^2 (0.22 and 0.30), relatively high AIC values (90.23 and 81.94), low AUC values, and comparably low classification accuracies (53.8% and 47.5%) (Table 3). However, compared to the contrast and orderliness groups, the univariate model using GLCM standard deviation achieved comparable (albeit slightly worse) values for McFadden R^2, AIC, AUC, and classification accuracy (Table 3).

The low performance of the GLCM mean in discriminating between damaged and undamaged buildings is logical. This texture measure describes the mean pixel value (considering the rescaled values) within the window. Pixel values are weighted based on frequency of occurrence in combination with their neighbors (Hall-Beyer, 2017a). Damaged buildings with low contrast/low order and undamaged buildings with high contrast/high order may have similar GLCM means, since the pixel values are weighted based on occurrence, and both classes contain the whole range of rescaled grey levels.

The low performance of GLCM correlation for discriminating between damaged and undamaged buildings is also logical. GLCM correlation describes the linear dependency, or spatial autocorrelation, of pixel values within a window (Hall-Beyer, 2017a). GLCM correlation is highly dependent on the sizes of objects or 'patches' within a window (Hall-Beyer, 2017a). The sizes of patches or range of sizes may be similar between the damaged and undamaged buildings. The estimated ranges from the damaged building variograms ranged from 0.5 to 1.0 m (Figure 7), and the ranges of the undamaged building variograms may be similar. Relative to the contrast and orderliness textures, the similar performance of GLCM standard deviation is logical since this measure describes the dispersion of the pixel values around the GLCM mean, and is similar to contrast and dissimilarity (Hall-Beyer, 2016a).

5.2 The effects of window size

The estimated ranges of the sample damaged building variograms ranged from 0.5 to 1.0 m (Figure 7). Therefore, the window size for texture calculation was greater than the maximum variogram range, at 51 x 51 pixels (1.4 x 1.4 m). A larger window size of 101 x 101 pixels (2.7 x 2.7 m) was also tested to compare damage classification accuracy. The use of the larger window for texture calculation resulted in the same classification accuracy for models #1 through #6, and lower accuracy for models #7 through #10 (Table 3). The results suggest that the 51 x 51 pixel window was adequately large for capturing the texture. Smaller windows are preferred over larger windows due to computation times.

5.3 Future work

5.3.1 Model validation with more diverse test data

The classification accuracies of the models in this study were calculated using a testing set containing 40 damaged buildings and 40 undamaged buildings whose roofs are >90% missing/destroyed and intact, respectively. Further classification accuracy testing should be performed on testing samples outside the study area. The UAV orthomosaics cover more than 20 km^2 of Sint Maarten (Figure 2), so randomly selected buildings from each orthomosaic should be included as testing samples. Additionally, buildings from other countries should also be included as testing samples, as building materials and construction types can vary in different parts of the world.

In addition to the 160 sample buildings used in this study, 300 buildings footprints were digitized. Most of these buildings have partially missing roofs. These buildings should be used as testing samples to see if there is a correlation between conditional probabilities calculated by the logistic regression models, and the percentage of missing roof.

5.3.2 Refinement of GLCM texture extraction

Future work on variogram analysis can include fitting spherical models to the sample variograms to provide more accurate estimates of variogram ranges. Additional improvements to variograms may include using a larger lag size to bring the nugget closer to zero, as the nuggets in the final variograms in this study were negative (Figure 7).

Two window sizes – 51 x 51 pixels (1.4 x 1.4 m) and 101 x 101 pixels (2.7 x 2.7 m) were compared for classification accuracy. The larger window size resulted in the same classification accuracy for 6 of 10 logistic regression models, and worse classification accuracy for 4 of 10 models (Table 3). Future window size testing can compare classification accuracy of windows larger than the maximum variogram range (1.0 m), but smaller than 51 pixels (1.4 m) to see the resulting classification accuracy. This has implications for computation time, as smaller windows require less time.

As computation time is decreased by using fewer grey levels, smaller windows sizes, and calculating fewer texture measures, computation time is also decreased by considering fewer neighbors relative to the reference pixel. In this study, texture was calculated omnidirectionally, where a reference pixel was compared against its 8 immediate neighbors, and the 8 values were averaged. Future work should test whether textures calculated with directionality (i.e., considering one neighbor only) yield similar results, or if omnidirectionality is necessary.

5.3.3 Exploration of Random Forest and other classification techniques

The univariate Random Forest models resulted in 12.5 to 27.5% classification accuracy increases over their logistic regression equivalents (Table 3). Although Random Forest classification was not the focus of this project, future work should investigate Random Forest classification more fully. This includes following the best practices regarding sample design as outlined by Millard and Richardson

(2015), such as working toward minimal spatial autocorrelation in the training and testing samples. Additional classification methods should also be explored.

5.3.4 Testing coarser image resolutions

The variogram analysis failed to determine a maximum suitable pixel size for this application, since Gaussian models could not be fitted to the sample variograms. An alternative approach can be to downsample the original 0.03 m input image to resolutions ranging from 0.10 to 0.50 meters, by increments of 0.05 m. Then, texture images can be calculated from the coarser images, and logistic regression models can be generated and compared for classification accuracy. However, it is unclear if downsampling the imagery would be accurately simulating image acquisition at higher altitudes. If appropriate, this analysis would help illuminate if the spatial resolution afforded by UAV-derived imagery (< 0.05 m) results in a significant improvement in the prediction of roof damage over spatial resolutions achievable by piloted aircraft and satellites. The maximum suitable pixel size for this application also has implications on the computation time. For example, it took 3.5 hours to generate the 8 texture images in PCI Geomatica using a high-performance computer (Intel® Core™ i7-5820K CPU @ 3.30 GHz with 32 GB RAM). Generating texture images using coarser input imagery would reduce computation time.

5.4 Limitations

The major limitations of this application are that: (i) a geospatially accurate building footprint vector file must pre-exist for the affected region, and (ii) the post-event imagery must also be geospatially accurate. The footprint vector file and imagery must align well in order to accurately extract the texture values of pixels pertaining to buildings only. Fixing alignment issues between the two

datasets is possible, but will increase the time spent on analysis. For densely populated regions that are vulnerable to natural disasters, it is recommended that a high-quality building footprint file is readily available. Alternatively, automated building segmentation approaches such as object-based image analysis (OBIA) may be considered, where texture calculation is already part of the image object segmentation process.

6. Conclusions

This project used GLCM textures as predictors of building roof damage. A post-Hurricane Irma UAV orthomosaic red band image was used to extract GLCM texture measures from 80 buildings with missing/damaged roofs (40 training and 40 testing samples), and 80 buildings with intact roofs (40 training and 40 testing samples). A variogram analysis provided information about spatial autocorrelation of pixel values within sample damaged buildings, and guided the decision about window size for texture image calculation. Eight GLCM texture images were generated using a 51 x 51 pixel window: contrast, dissimilarity, homogeneity, ASM, entropy, GLCM mean, GLCM standard deviation, and GLCM correlation. The mean texture values within each building were used to represent the building. The mean texture values were used as candidate predictors for logistic regression model building.

The logistic regression model with the best overall goodness-of-fit statistics and classification accuracy (71.3%) was the univariate model containing mean ASM as the predictor. The remaining logistic regression model classification accuracies ranged from 47.5 to 70.0%. It is important to note that the two multivariate models (containing two and three predictors) did not result in higher classification accuracies than the univariate models. The univariate models that had the highest classification accuracies contained the orderliness contrast textures (ASM, entropy) as predictors (70.0 to 71.3%

accuracies), followed by models that had contrast textures (contrast, dissimilarity, homogeneity) as predictors (67.5 to 68.8% accuracies). The descriptive textures generally performed more poorly.

A second set of texture images was generated with a 101 x 101 pixel window to compare the classification accuracies of the resulting models. The models generated with the larger (101 x 101 pixel) window resulted in the same classification accuracies for 6 of 10 models, and lower classification accuracies for the remaining 4 models, when compared to the models generated with the smaller (51 x 51 pixel) window. These results suggest that 51 x 51 pixels (1.4 x 1.4 m) is an adequately large window size. This window size was determined by estimating the ranges of variograms generated for 10 sample damaged buildings, and setting the window size (1.4 x 1.4 m) slightly larger than the maximum variogram range (1.0 m).

The predictors generated using the 51 x 51 window for texture calculation were also used in Random Forest classification. Considering the univariate models only, the Random Forest models had classification accuracies that were 12.5 to 27.5% higher than their logistic regression equivalents. Given the high accuracies, future work will focus on understanding the Random Forest algorithm more deeply, as well as other machine learning algorithms.

Other future work will include downsampling the 0.03 m UAV orthomosaic in order to test the effect of coarser image resolution on the classification accuracy of building damage using GLCM texture. This has important implications for suitable imaging platforms for this application. Covering as much area as possible is key in a disaster response scenario, so piloted aircraft and satellites are preferred, but the potential differences in damage classification accuracy are not well understood.

7. References

Ajmar, A., Boccardo, P., Disabato, F., & Giulio Tonolo, F. (2015). Rapid Mapping: geomatics role and research opportunities. Rendiconti Lincei, 26, 63–73. http://doi.org/10.1007/s12210-015-0410-9

Akar, & Güngör, O. (2015). Integrating multiple texture methods and NDVI to the Random Forest classification algorithm to detect tea and hazelnut plantation areas in northeast Turkey. International Journal of Remote Sensing, 36(2), 442–464. https://doi.org/10.1080/01431161.2014.995276

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

CEMS. (2017). EMSR234: Hurricane Irma in Sint Maarten. Retrieved from: http://emergency.copernicus.eu/mapping/list-of-components/EMSR234

CEMS. (2018). EMS – Rapid Mapping products. Retrieved from: http://emergency.copernicus.eu/mapping/ems/ems-rapid-mapping-products

Chen, J., Liu, H., Zheng, J., Lv, M., Yan, B., Hu, X., & Gao, Y. (2016). Damage Degree Evaluation of Earthquake Area Using UAV Aerial Image. International Journal of Aerospace Engineering, 2016. https://doi.org/10.1155/2016/2052603

Dell'Acqua, F., & Gamba, P. (2012). Remote sensing and earthquake damage assessment: Experiences, limits, and perspectives. Proceedings of the IEEE, 100(10), 2876–2890. https://doi.org/10.1109/JPROC.2012.2196404

Denis, G., de Boissezon, H., Hosford, S., Pasco, X., Montfort, B., & Ranera, F. (2016). The evolution of Earth Observation satellites in Europe and its impact on the performance of emergency response services. Acta Astronautica, 127, 619–633. http://doi.org/10.1016/j.actaastro.2016.06.012

Dong, L., & Shan, J. (2013). A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. ISPRS Journal of Photogrammetry and Remote Sensing, 84, 85–99. https://doi.org/10.1016/j.isprsjprs.2013.06.011

Fernandez Galarreta, J., Kerle, N., & Gerke, M. (2015). UAV-based urban structural damage assessment using object-based image analysis and semantic reasoning. Natural Hazards and Earth System Sciences, 15(6), 1087–1101. http://doi.org/10.5194/nhess-15-1087-2015

Geldsetzer, T. (2018a). GEOG 639 (W2018): Lecture 10 [PowerPoint presentation]. Retrieved from Desire2Learn.

Geldsetzer, T. (2018b). GEOG 639 (W2018): Lecture 4 [PowerPoint presentation]. Retrieved from Desire2Learn.

Geldsetzer, T. (2018c). GEOG 639 (W2018): Lecture 2 [PowerPoint presentation]. Retrieved from Desire2Learn.

Hall-Beyer, M. (2017a). GLCM Texture: A Tutorial v. 3.0 March 2017. Retrieved from: https://prism.ucalgary.ca/handle/1880/51900

Hall-Beyer, M. (2017b). Practical guidelines for choosing GLCM textures to use in landscape classification tasks over a range of moderate spatial scales. International Journal of Remote Sensing, 38(5), 1312–1338. http://doi.org/10.1080/01431161.2016.1278314

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. IEEE Transactions on Systems, Man and Cybernetics 3: 610–621. doi:10.1109/TSMC.1973.4309314.

Haralick, R. M. (1979). Statistical and Structural Approaches to Texture. Proceedings of the IEEE 67:786–804.

Liu, B., & Liew, S. C. (2007). Texture retrieval using grey-level co-occurrence matrix for Ikonos panchromatic images of earthquake in Java 2006. International Geoscience and Remote Sensing Symposium (IGARSS), 286–289. https://doi.org/10.1109/IGARSS.2007.4422786

Millard, K., & Richardson, M. (2015). On the importance of training data sample selection in Random Forest image classification: A case study in peatland ecosystem mapping. Remote Sensing, 7(7), 8489–8515. https://doi.org/10.3390/rs70708489

PCI Geomatics. (2017). TEX. Retrieved from: http://www.pcigeomatics.com/geomatica-help/references/pciFunction_r/python/P_tex.html

Plank, S. (2014). Rapid damage assessment by means of multi-temporal SAR-A comprehensive review and outlook to Sentinel-1. Remote Sensing (Vol. 6). https://doi.org/10.3390/rs6064870

R Core Team. (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rogerson, P.A., (2006). Statistical Methods for Geography. London: Sage Publications Ltd

Stoltzfus, J. C. (2011). Logistic regression: A brief primer. Academic Emergency Medicine, 18(10), 1099–1104. http://doi.org/10.1111/j.1553-2712.2011.01185.x

Sui, H., Tu, J., Song, Z., Chen, G., & Li, Q. (2014). A Novel 3D Building Damage Detection Method Using Multiple Overlapping UAV Images. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-7(October), 173–179. https://doi.org/10.5194/isprsarchives-XL-7-173-2014

Vetrivel, A., Gerke, M., Kerle, N., & Vosselman, G. (2015). Identification of damage in buildings based on gaps in 3D point clouds from very high resolution oblique airborne images. ISPRS Journal of Photogrammetry and Remote Sensing, 105, 61–78. http://doi.org/10.1016/j.isprsjprs.2015.03.016