# Learning From Synthetic Models for Roof Style Classification in Point Clouds

Xi Zhang
Illinois Institute of Technology
Chicago, IL 60616
xzhang22@hawk.iit.edu

Andi Zang
Illinois Institute of Technology
Chicago, IL 60616
azang@hawk.iit.edu

Gady Agam
Illinois Institute of Technology
Chicago, IL 60616
agam@iit.edu

Xin Chen
NOKIA Here.com
Chicago, IL 60606
xin.5.chen@here.com

## ABSTRACT

Automatic roof style classification using point clouds is useful and can be used as a prior knowledge in various applications, such as the construction of 3D models of real-world buildings. Previous classification approaches usually employ heuristic rules to recognize roof style and are limited to a few roof styles. In this paper, the recognition of roof style is done by a roof style classifier which is trained based on bag of words features extracted from a point cloud. In the computation of bag of words features, a key challenge is the generation of the codebook. Unsupervised learning is often misguided easily by the data and detects uninteresting patterns within the data. In contrast, we propose to integrate existing knowledge of roof structure and cluster the points of target roof styles into several semantic classes which can then be used as code words in the bag of words model. We use synthetic variants of these code words to train a semantics point classifier. We evaluate our approach on two datasets with different levels of degradations. We compare the results of our approach with two unsupervised learning algorithms: K-Means and Gaussian Mixture Model. We show that our approach achieve higher accuracy in classification of the roof styles and maintains consistent performance among different datasets.

## Categories and Subject Descriptors

I.4.8 [**Image Processing And Computer Vision**]: Scene Analysis-Range data

## General Terms

Algorithm, Experimentation, Performance

## Keywords

Point Cloud, Roof Style, Classification, Machine Learning

## 1. INTRODUCTION

Realistic roof modelling has become increasingly important for urban planning, navigation and GIS. There are two main approaches: reconstructing a mesh from real point cloud data and artificially designing a mesh model. Reconstructing meshes from raw point cloud data [13, 18, 19, 11] preserves detailed building geometry and is suitable for modeling the core area of a city. However, without given any prior knowledge about building's roof style, many geometric and topological constraints are used by these methods, which in turn make the methods hard to be applied to a different dataset. Artificial meshes on the other hand is broadly used and an ideal solution for modeling buildings with simple shape. The workload and cost of the artificial modeling will be largely reduced if roof style and other properties of the building are given beforehand.

Therefore it is valuable to recognize before modeling the roof style of a given building data. Existing works on roof recognition identify roof styles by using pre-defined rules to look for specific features of the different roof styles. Such previous methods are restricted to very few roof styles. In this paper, we propose a learning based roof style classification algorithm using aerial LiDAR point clouds. The proposed approach is tested in classifying 9 roof styles and to our knowledge, it is the first method that is not limited to a small set of styles. In contrast to the previous work, the proposed approach does not depend on geometry fitting and roof topology detection. Instead, our approach to classifying roof style is based on a bag of words of roof point semantics.

The novel contributions of this paper are in several aspects. First, we propose a learning based roof style classification algorithm using a point cloud data. The proposed approach capable of recognizing 9 different roof styles. Second, our learning based approach is the first to successfully apply the bag of words model for the classification of the roof styles. We show that by integrating existing knowledge about the roof structures, it is possible to obtain meaningful code words that support good recognition performance. We test our approach on multiple data sets with different types and levels of degradation and compared it to known methods. We show that our approach achieves better accuracy and keep consistent performance over several datasets.

## 2. RELATED WORK

The major difficulties in roof style modelling using point cloud data is to deal with the presence of outliers and missing parts in the data.

*SIGSPATIAL'14,* November 04 - 07 2014, Dallas/Fort Worth, TX, USA
Copyright 2014 ACM 978-1-4503-3131-9/14/11 ...$15.00
http://dx.doi.org/10.1145/2666310.2666407

The most common approach to address the noise problems to fit a geometric surfaces to the point cloud. For example, in [17], a 3D Hough transform is used to fit several planes on the roof. To prevent from over fitting, roof patches are divided and fit by planes separately. A similar strategy of fitting plane has been used in [9], where the authors employ in addition several heuristic rules to detect roof ridges. To understand adjacency relationship among roof pieces, a topology graph is proposed in [16]. This was then used by many others [7, 6, 9]. Roof detection and classification based on learning algorithms have been used in several cases. In [15], to help detect gable and the hip roofs, height information is used to separate these two roof styles using a linear classifier. In [5], by measuring the linearity of roof contours, a neural network is constructed to separate a building region from everything else. To better deal with residual sensor noise and small features, [18] extent classic dual contouring method in their approach. By minimizing a quadratic error function, their approach can handle reconstruction simplification as easy as optimizing the error function. In [13], refined boundaries of building region are used to generate building facades, thus better support the modeling of building roofs. A strategy of using combination of 3D-primitive and mesh-patches to represent building roofs is employed in [11], which better controls the error of mesh simplification. To correct local fitting error and keep better global consistency, a modeling method which look in to global regularities is introduced in [19]. By discovering a various of global constraints, their method gains a large improvement on both quantitative and qualitative evaluation.

Existing work for roof style classification suffers from low classification accuracy, due to the use of heuristic rules and the assumption of specific geometry of roof data. In contrast, the proposed approach does not make any specific assumption about the underlying geometry of the roof styles. We propose to use the bag of words model in classifying the roof styles. Because bag of words features represent an object in a statistical manner, it can better handle outliers in the data, and hence has better classification results. The use of the bag of words model in this work is based on treating each roof as an entity composed of different semantic parts. Several examples of semantic roof points in a hip roof are shown in Figure 1.

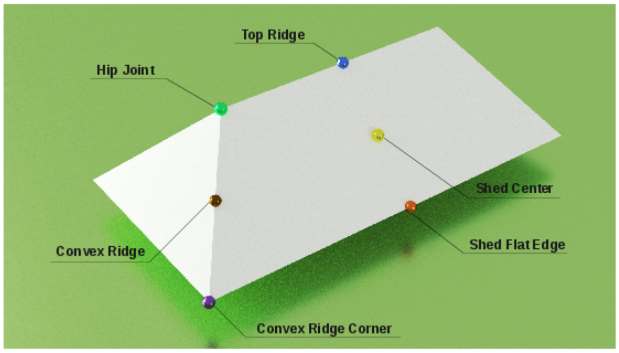


**Figure 1: Illustration of the point semantics in a hip roof.**

A key step in generating bag of words features is the generation of a highly descriptive codebook. Usually, the codebook is generated using an unsupervised learning algorithms such as K-Means or Gaussian Mixture Model. However, such unsupervised algorithms are often very hard to tune and need a large amount of fine tuning to generate a meaningful and stable model. Hence, instead of learning a codebook from sample data, we design a codebook containing 33 kinds of semantic roof points. These are shown in Figure 13. Our approach is currently capable of recognizes 9 roof styles including Curve, Flat, Gable, Hex, Hip, Mansard, Pyramid, Shed and Unknown. Our approach can easily accommodates new roof styles by re-training of the new dataset with the new roof styles added.

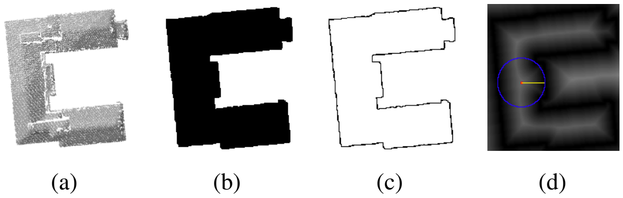


**Figure 2: An example of computing the largest inscribed circle for a U-shaped roof. (a) The original roof points. (b) The footprint image of roof mapped down to 2D plane. (c) The image of building footprint edges extracted from (b). (d) The inscribed circle of distance transform image of (c)**

## 3. ROOF PREPROCESSING

In our work, the roof is processed automatically one at a time. To extract point data associated to a roof from aerial LiDAR, footprints of buildings are first generated using a method described in [13] and then are used to extract roof points. The proposed roof style classification algorithm is invariant to size and resolution of the point cloud due to pre-processing of data. In our approach, we assume that the fraction of a specific kind of semantic roof points in a roof should remain consistent in roofs with the same style. To achieve this consistency among different roofs, we need to deal with two pre-processing problems: roof size normalization and point cloud re-sampling.

**•Normalization**

We normalize all the roofs to a uniform size. We define the size of roof as the radius of the largest inscribed circle of a roof's footprint. To get the radius of the inscribed circle, we first project all roof points to a 2D binary footprint image. Small holes of the footprint image are then filled using a morphology operation. To get inscribed circle, we extract edges of the footprint image and compute the distance transform[2] for the edge image. We identify the center of the largest inscribed circle as the pixel with the largest distance within the roof (see Figure 2). In our experiments, the normalization scales the synthetic and actual roof models to have a largest inscribed circle with a diameter of 5 meters.

**•Re-sampling**

The re-sampling in our approach is done by first up-sampling and then down-sampling the point cloud to meet a pre-defined resolution. To compute the average distance $\Theta$ in the current mesh, we compute the distance between each point and its closest neighbor and average this distance. We then try to sampling the point cloud to match this distance to a specified distance of $\Theta^\star$ (0.1 meters in our experiments).

The proposed re-sampling algorithm first generates a denser version of the roof point cloud by up-sampling the roof data. Subsequently, the point cloud is down-sampled to meet the required resolution. This is done to assure we can achieve the desired resolution even if the distribution of points in the original data set is not uniform. The up-sampling is based on subdivision of a triangle mesh reconstructed from the raw point cloud. To reconstruct

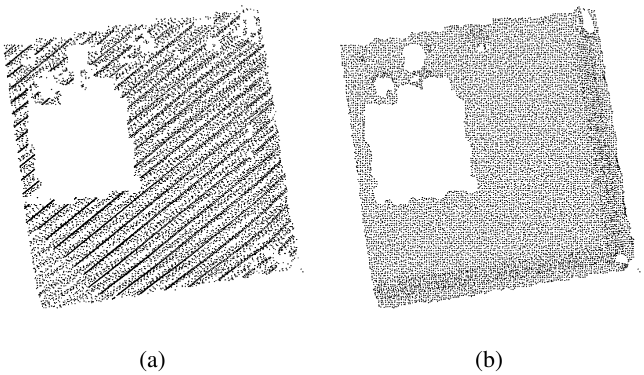


**Figure 3: Demonstration of our re-sampling approach. (a) The point cloud of the original roof. (b) The re-sampled point cloud produced by the proposed approach.**

the mesh, we use the Ball-Pivoting Algorithm (BPA) [1] and set the radius of the ball to $\gamma = 1.5\Theta$. We then employ the midpoint subdivision algorithm to subdivide edges of the mesh whose length is longer than $\Theta^\star$ to produce a denser roof mesh. To down-sample the up-sampled mesh, we use an octree, and set the size of the leaf node to $\Theta^\star$. In each leaf node, we retain the point closest to cell center. This leads to a down-sampled point cloud with the required resolution. An example of roof data before and after re-sampling is shown in Figure 3.

# 4. ROOF POINT SEMANTICS QUANTIZATION

## 4.1 Codebook and Synthetic Variants

The objective here is to produce a codebook of important roof points. This codebook can then be used for bag of words recognition. Different from the known approaches, we do not learn the codebook from the real dataset. We instead manually represent the codes in the codebook as semantic parts of the roof structure produced by manually analysing roof style models. We then learn a point semantics classifier using a large amount of synthetic variants of roof styles. There are 33 codes in the codebook used in this work. The key points that are used to produce the code words are shown in Figure 13.

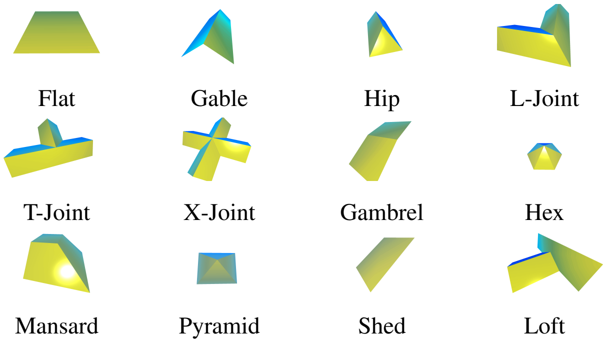


**Figure 4: Illustration of all roof base models that are used in this work.**

As can be observed in Figure 13, some roof styles shown in Figure 4 are used to produce multiple key points. The synthetic variants of the codes are derived from these base models in two ways. First, we generate roof models by changing the parameters of the roofs including height, width and slope. The parameters are drawn evenly within a given range. Second. we consider erosions of roofs because it is common in our data to have eroded roofs due to unknown reasons. Four versions with different magnitude of erosion are derived using approach described in [8] from each generated roof. Please refer to appendix section for more information regarding creation of the synthetic variants.

## 4.2 Point Descriptor

To have distinct code words for each of the key points we define in the synthetic model, we need to have good characterization of the points that take into account their neighborhood. Without such characterization, classification of code words in actual data may be inaccurate. The point descriptors we use are composed of two groups of features: spatial features and contextual features. The point classifier has two stages. The first stage uses spatial feature whereas the second stage uses contextual features.

### • Spatial Features

Spatial features take into account the neighborhood of each point. So the spatial features are powerful to describe geometry characteristics in the point's neighborhood and so assist in producing a distinct characterization of it. Let $p_i$ be a key point. Let $N_i$ be a neighborhood of the point, containing neighbors in a radius of $\mu$ (0.6 meters in our experiments). We denote the spatial features at $p_i$ using $\mathcal{S}_i$.

**i. Eigen Features (EF).** We compute the eigenvalues: $\lambda_1, \lambda_2, \lambda_3$ of the covariance matrix of neighbors $N_i$ centred at $p_i$. We then add following features for $p_i$: $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_3 - \lambda_2$, $\lambda_2 - \lambda_1$, $\lambda_1/(\lambda_1+\lambda_2+\lambda_3)$, $\lambda_2/(\lambda_1+\lambda_2+\lambda_3)$, $\lambda_3/(\lambda_1+\lambda_2+\lambda_3)$, yielding 8 features in total. These features capture the non-planarity of points around $p_i$.

**ii. Point Feature Histogram (PFH).** These are point cloud features based on histogram which are described in [14]. Taking into account computational efficiency, 3 subdivisions of the features range are used in the feature histogram which yield $3^3 = 27$ features.

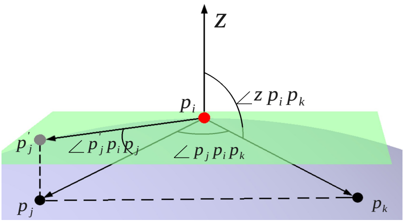


**Figure 6: Illustration of the frame we used in the computation of the shape distribution features.**

**iii. Shape Distribution Features (SD).** Shape distribution [12] measures global geometric properties of an object by representing object features as a probability distribution. Shape distribution is invariant to translation, rotation and scale and is highly informative in matching objects. Using the ideas of shape distribution we construct four features:

**1) A2:** Measures the angle between two vectors composed by two neighbors $p_j, p_k \in N_i$ and $p_i$, shown as $\angle p_j p_i p_k$ in Figure 6. A2 feature is computed for all pairs of points chosen from $N_i$.

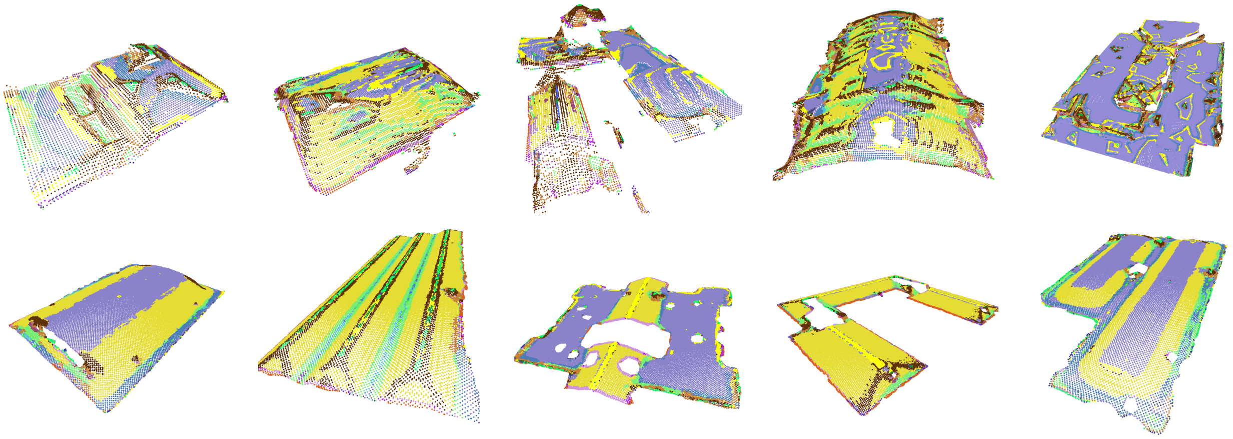


**Figure 5: Point classification results obtained by running our point type classifier on dataset one (top row) and the dataset two (bottom row). The colors of points corresponds to the color bars of the point in Figure 13.**

**2) Az:** Measures the angle between the z direction and a vector pointing from $p_i$ to one neighbor $p_k$, shown as $\angle z p_i p_k$ in Figure 6. All points in $N_i$ are used to compute this feature.

**3) D2:** The is the feature D2 as described in [12]. It measures the distance between any two neighbors of $p_i$, an example is shown as $\|p_j p_k\|$ in Figure 6. D2 is computed for all pairs of points chosen from $N_i$.

**4) Dt:** Measures the angle between $p_i$'s tangent plane and a vector pointing from $p_i$ to another neighbor $p_j$, shown as $\angle p'_j p_i p_j$ in Figure 6, where $p'_j$ is the projection of $p_j$ on tangent plane. All points in $N_i$ are used to compute this feature.

We use a histogram with 10 bins to represent each of the features described above. In total there are 40 SD features that are computed.

**iv. Spin Image.** Spinning around the $z$ direction, we compute a spin image [10] with $6(width) \times 11(height)$ dimensions. Totally 66 features are contributed by spin image.

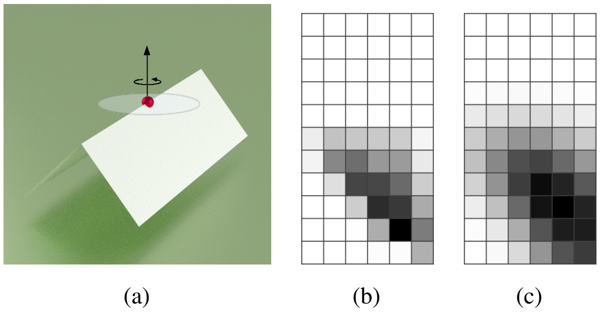


**Figure 7: Example of applying Gaussian smoothing to the spin image features. (a) The red dot shows the location where spin image features are computed. (b) Generated spin image without smoothing. (c) Result of spin image after smoothing.**

To better handle variations, noises and outliers in the actual dataset, since training is done on a synthetic set, we smooth all above features except for the EF features. Specifically, we spread the value in each entry of the feature vector to its adjacent entries using Gaussian convolution. A $3 \times 3$ 2D Gaussian kernel is applied to the spin images, and a size 3 1D Gaussian kernel is applied to the PFH and SD features. An example of the spin image after convolution is shown in Figure 7.

We use a random forest to learn the first stage of the point classifier based on spatial features. The classifier produces a probability measure for each of the 33 code words. Thus given a point $p_i$ we have $\{P(l_j|\mathcal{S}_i)\}_{j=1}^{33}$, where $\mathcal{S}_i$ is the spatial feature set at $p_i$ and $l_j$ is the $j$-th label ($j \in [1, 33]$).

### • Contextual Features

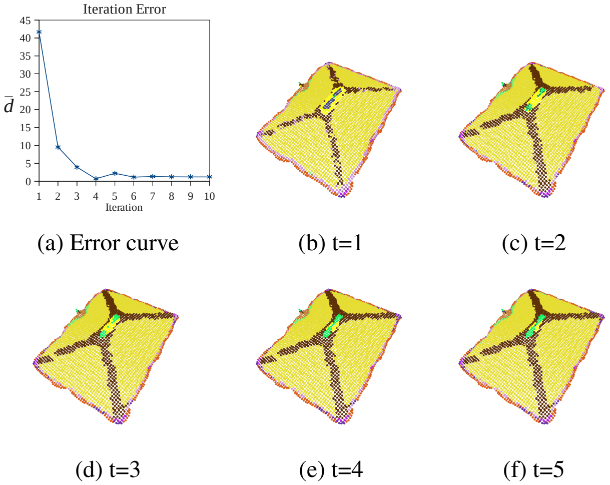
The point classifier in the previous stage attempts to classify a single point. More reliable results can be obtained if considering the labels of neighbors as features. However, since initially the labels of a point or its neighbors are not known, an iterative process is required. For each point $p_i$, let $\mathcal{C}_i^t,\ t \geq 1$ represent the contextual features of point $p_i$ in $t$-th iteration which are composed of assignment values and let $\mathcal{C}_i^t[j]$ to denote $j$-th element in $\mathcal{C}_i^t$. The second level classifier starts with an assignment vector $\mathcal{C}_i^1$ containing the assignment value of point $p_i$ into all 33 code words. This assignment is a sum of the assignment at $p_i$ and all of its neighbors $\{p_k|p_k \in N_i\}$. The second level classifier then uses $\mathcal{C}_i^1$ and $\mathcal{S}_i$ to regenerate the assignment. In each iteration we update the $\mathcal{C}_i^t[j]$ using the following rule:

$$\mathcal{C}_i^t[j] = \begin{cases} \sum\limits_{p_k \in N_i} P(l_j \mid \mathcal{S}_k) & \text{if} \quad t = 1 \\ \sum\limits_{p_k \in N_i} P(l_j \mid \mathcal{S}_k, \mathcal{C}_k^{t-1}) & \text{if} \quad t \geq 2 \end{cases}, \qquad (1)$$

## 4.3 Point Semantics Quantization

In both stages of code word classifier, we use a random forest[3]. There are two parameters in setting the random forest: the number of trees $\rho_0$ in the forest and the number of features $\rho_1$ that the random forest can choose at each node. Assuming that the dimension of the input feature vector is $\epsilon$, so we set $\rho_0 = 100$ and $\rho_1 = \sqrt{\epsilon}$ for both the random forests.

Training and testing of the first classifier is straight forward. For the second classifier, both training and testing are iterative. To determine the convergence of the classification, In $t$-th iteration, we use



(a) Error curve (b) t=1 (c) t=2

(d) t=3 (e) t=4 (f) t=5

**Figure 8: Example of iteratively refining the code word classification using the contextual features. (a) A curve of the corresponding $\bar{d}$ in each iteration. (b) to (f) The results of the classification in each iteration. Points are colored using the highest probability label.**

a histogram $\mathcal{H}^t = \{h_j^t\}_{j=1}^{33}$ to count the frequency of the points in a roof in $t$-th iteration. Given points of a roof $\mathcal{P} = \{p_i\}_{i=1}^{n}$, each bin $h_j^t$ of $\mathcal{H}^t$ is computed as:

$$h_j^t = \frac{1}{n}\sum_{i=1}^{n} P(l_j \mid \mathcal{S}_i, \mathcal{C}_i^{t-1}), \quad 1 \le j \le 33 \tag{2}$$

Convergence is determined by comparing the difference of histograms in two continuous iterations using the following equation:

$$\bar{d} = \sum_{j=1}^{33} \frac{|h_j^t - h_j^{t-1}|}{33} \tag{3}$$

The algorithm is determined to converge when $\bar{d} \le \xi$, ($\xi = 2\%$ number of roof points in our experiments). An example of the classification of roof points by the proposed approach is shown in Figure 8. As can be observed in this figure, context features produce more regular and confident result. After convergence, the histogram $\mathcal{H}$ is used as the bag of words features of the roof.

# 5. ROOF STYLE CLASSIFIER

The roof style classifier uses the bag of words features $\mathcal{H}$. We use a random forest classifier to classify each roof style into one of 9 possible roof styles. To accommodate various kinds of point cloud degradations in different datasets, the roof style classifier is trained using a real roof dataset. We set the two parameters of the random forest as: $\rho_0 = 100$ and $\rho_1 = 6$. To avoid bias towards a particular roof style and create a balanced number of training data among different styles of the roofs, we super sample the training data using the SMOTE [4] algorithm.

# 6. EXPERIMENTAL RESULTS

The proposed approach has been tested on two datasets. In the first dataset, there are 3290 buildings that were extracted from Chicago urban area. In the second dataset there are 3290 buildings that were extracted from a San Francisco urban area. The two datasets have different characteristics and kinds of degradations.

| | Chicago | | | | | | San Francisco | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Precision | | | Recall | | | Precision | | | Recall | | |
| | GMM | KM | Ours | GMM | KM | Ours | GMM | KM | Ours | GMM | KM | Ours |
| FLAT | 0.87 | 0.85 | 0.92 | 0.88 | 0.90 | 0.90 | 0.88 | 0.88 | 0.86 | 0.93 | 0.93 | 0.93 |
| SHED | 0.94 | 0.94 | 0.91 | 0.57 | 0.64 | 0.78 | 0.87 | 0.91 | 1.00 | 0.43 | 0.68 | 0.68 |
| GABLE | 0.62 | 0.67 | 0.71 | 0.86 | 0.84 | 0.88 | 0.57 | 0.61 | 0.65 | 0.71 | 0.69 | 0.77 |
| HIP | 0.65 | 0.63 | 0.63 | 0.16 | 0.36 | 0.37 | 0.55 | 0.61 | 0.70 | 0.22 | 0.28 | 0.31 |
| HEX | 0.87 | 0.86 | 0.93 | 0.87 | 0.81 | 0.90 | 0.90 | 0.80 | 0.92 | 0.83 | 0.66 | 1.00 |
| PYRAMID | 0.83 | 0.66 | 1.00 | 0.20 | 0.16 | 0.37 | 0.87 | 0.83 | 1.00 | 0.87 | 0.93 | 1.00 |
| MANSARD | 1.00 | 0.75 | 1.00 | 0.25 | 0.18 | 0.31 | 0.50 | 0.66 | 1.00 | 0.05 | 0.11 | 0.41 |
| CURVED | 1.00 | 0.93 | 1.00 | 0.87 | 0.93 | 1.00 | 0.71 | 0.70 | 0.74 | 0.77 | 0.71 | 0.79 |
| UNKNOWN | 0.84 | 0.85 | 0.97 | 0.88 | 0.85 | 0.90 | 0.62 | 0.59 | 0.66 | 0.63 | 0.64 | 0.66 |
| Average | 0.85 | 0.79 | 0.89 | 0.62 | 0.63 | 0.71 | 0.72 | 0.73 | 0.84 | 0.60 | 0.63 | 0.73 |

**Table 1: Precision and recall of Gaussian Mixture Model(GMM), K-Means(KM) and our approach are shown in above table. We use red font to show highest value among results obtained by these three approaches.**

In the first dataset, roof points are irregular and the shape of the roof is decayed to some extent. This is due to the fact that roofs in this dataset were originally produced from a highly down-sampled aerial LiDAR. The roof points in the second dataset have relatively low resolution and uneven distribution across the roof surface. Both datasets contain roof points of building only. The roofs were labelled to one of 9 target roof styles according to their appearances. A roof is labelled as UNKNOWN when either the roof is not recognizable or its roof style can not be categorized into one of the 8 styles. When a roof is composed of multiple styles, we label the roof according to the style of the largest component. The distribution of roof styles in the two datasets is shown in Figure 9.

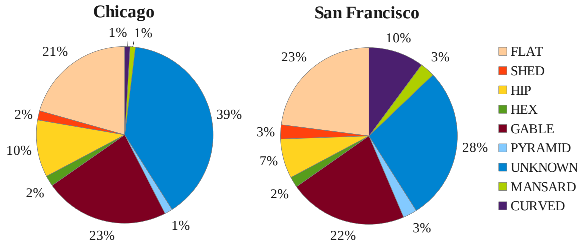


**Figure 9: The distribution of the roof styles in the two datasets.**

To evaluate the performance of our approach and measure the improvement in using the synthetic model when generating the codebook, we compare the roof style classification results obtained by our approach to the results obtained by the K-Means (KM) and the Gaussian Mixture Model (GMM) algorithms for generating the codebook.

We evenly divide the dataset into two parts, one part is used as a training set, while the second part is used as a testing set. In the case of the KM and GMM algorithms, the codebook is generated using the training set. For each approach, the bag of words features of all buildings are then computed using its own codebook. We set $k = 30$ in the K-Means algorithm and $k = 27$ in the Gaussian Mixture Model algorithm as these were the two configurations that provided the best accuracy for these approaches. The roof style classifier of each approach is then trained and tested using its own bag of words features. The results of these three approaches in terms of precision and recall are shown in Table 1. F-Score results are shown in Figure 10.

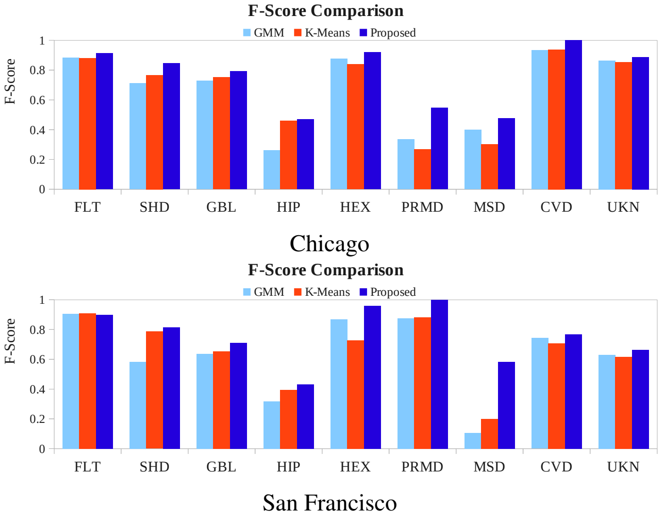
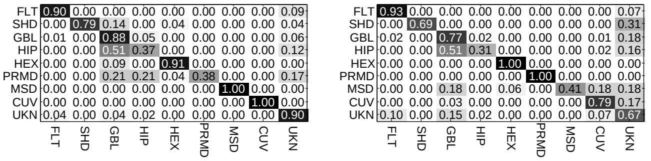


**Figure 10: The comparison of F-Score on the two datasets by running KM, GMM and the proposed approach.**

As can be observed, the proposed approach performs better for almost all roof styles. Considering the roof styles of PYRAMID and MANSARD, we observe that the performance of KM and GMM is limited by the fact that the training set does not contain many examples of such roofs. In contrast, the proposed approach uses synthetic models and so is not affected by a small set of training examples.

The confusion matrices of the proposed approach on the two datasets is shown in Figure 11. As can be observed, the proposed approach performs well in classifying most of the roof styles. Errors mostly come from the confusion between the hip and gable roof styles, which are actually similar. Indeed, visual examination confirms that in our dataset many hip roofs resemble gable roofs due to shape erosion.



| | FLT | SHD | GBL | HIP | HEX | PRMD | MSD | CUV | UKN |
|---|---|---|---|---|---|---|---|---|---|
| FLT | 0.90 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.09 |
| SHD | 0.00 | 0.79 | 0.14 | 0.00 | 0.04 | 0.00 | 0.00 | 0.00 | 0.04 |
| GBL | 0.01 | 0.00 | 0.88 | 0.05 | 0.00 | 0.00 | 0.00 | 0.00 | 0.06 |
| HIP | 0.00 | 0.00 | 0.51 | 0.37 | 0.00 | 0.00 | 0.00 | 0.00 | 0.12 |
| HEX | 0.00 | 0.00 | 0.09 | 0.00 | 0.91 | 0.00 | 0.00 | 0.00 | 0.00 |
| PRMD | 0.00 | 0.00 | 0.21 | 0.21 | 0.04 | 0.38 | 0.00 | 0.00 | 0.17 |
| MSD | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 |
| CUV | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 |
| UKN | 0.04 | 0.00 | 0.04 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 | 0.90 |

Chicago

| | FLT | SHD | GBL | HIP | HEX | PRMD | MSD | CUV | UKN |
|---|---|---|---|---|---|---|---|---|---|
| FLT | 0.93 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.07 |
| SHD | 0.00 | 0.69 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.31 |
| GBL | 0.02 | 0.00 | 0.77 | 0.02 | 0.00 | 0.00 | 0.00 | 0.01 | 0.18 |
| HIP | 0.00 | 0.00 | 0.51 | 0.31 | 0.00 | 0.00 | 0.00 | 0.02 | 0.16 |
| HEX | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| PRMD | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 |
| MSD | 0.00 | 0.00 | 0.18 | 0.00 | 0.06 | 0.00 | 0.41 | 0.18 | 0.18 |
| CUV | 0.00 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 0.79 | 0.17 |
| UKN | 0.10 | 0.00 | 0.15 | 0.02 | 0.00 | 0.00 | 0.00 | 0.07 | 0.67 |

San Francisco

**Figure 11: The confusion matrix of obtained by the proposed approach on two datasets.**

The hardest part in the classification is the recognition of the unknown roof styles, because there is no regular pattern (bag of words features) for roofs in this category. We observe that we get accuracy above 90% in the Chicago dataset and accuracy above 67% in the San Francisco dataset.

We evaluate the performance of the proposed approach as a function of the training set size. The F-Score obtained by the proposed approach as a function of training set size is shown in Figure 12.

As can be observed, the proposed approach achieves stable performance on most of the roof styles. For roof styles with relatively low score, the curves present are ascending which could suggest that a better score can be obtained once more training data is included.

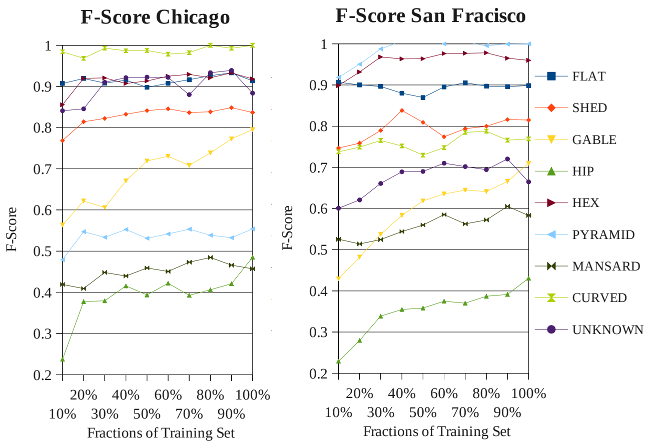


**Figure 12: The F-Score of each roof style obtained by using an increasing proportion of the training data.**

## 7. CONCLUSION AND FUTURE WORK

In this paper, we propose a learning based approach to perform roof style classification in aerial LiDAR. The advantages of the proposed approach lies in two aspects: first, the classification of roof style is based on recognition using the bag of words features which are composed of the classified roof points; second, we successfully generate the codebook based on synthetic models and knowledge about roof styles. We test our approach on two datasets with different characteristics, and compare the results we obtained to ones obtained using the K-Means and Gaussian Mixture Model algorithms. We show that our approach achieves better classification and keeps consistent performance in both datasets. In the future, we will look into how more sophisticated learning techniques such as deep learning and transfer learning can be applied and add more value to this work.

## 8. REFERENCES

[1] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):349–359, 1999.

[2] G. Borgefors. Distance transformations in digital images. *Computer vision, graphics, and image processing*, 34(3):344–371, 1986.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002.

[5] A. F. Elaksher and J. s. Bethel. Automatic generation of high-quality three-dimensional urban buildings from aerial images. *URISA Journal*, 20(1):5–13, 2008.

[6] S. O. Elberink and G. Vosselman. Building reconstruction by target based graph matching on incomplete laser data: Analysis and limitations. *Sensors*, 9:6101–6118, 2009.

[7] S. O. Elberink and G. Vosselman. Target graph matching for building reconstruction. *Laserscanning 2009; International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences:*, 38, 2009.

[8] G. Guennebaud and M. Gross. Algebraic point set surfaces. 26(3):23, 2007.

[9] H. Huang and C. Brenner. Rule-based roof plane detection and segmentation from laser point clouds. In *Urban Remote Sensing Event (JURSE), 2011 Joint*, pages 293–296. IEEE, 2011.

[10] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5):433–449, 1999.

[11] F. Lafarge and C. Mallet. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *International journal of computer vision*, 99(1):69–85, 2012.

[12] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *Graphics, ACM Transactions on*, 21(4):807–832, 2002.

[13] C. Poullis and S. You. Automatic reconstruction of cities from remote sensor data. In *Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Computer Society Conference on*, pages 2775–2782. IEEE, 2009.

[14] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Persistent point feature histograms for 3d point clouds. *IAS-10*, page 119, 2008.

[15] S. Valero, J. Chanussot, and P. Gueguen. Classification of basic roof types based on vhr optical data and digital elevation model. In *Geoscience and Remote Sensing Symposium (IGARSS), 2008 IEEE International*, volume 4, pages IV–149. IEEE, 2008.

[16] V. Verma, R. Kumar, and S. Hsu. 3d building detection and modeling from aerial lidar data. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Computer Society Conference on*, volume 2, pages 2213–2220. IEEE, 2006.

[17] G. Vosselman and S. Dijkman. 3d building model reconstruction from point clouds and ground plans. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4):37–44, 2001.

[18] Q.-Y. Zhou and U. Neumann. 2.5 d dual contouring: a robust approach to creating building models from aerial lidar point clouds. In *Computer Vision (ECCV), 2010 European Conference on*, pages 115–128. Springer, 2010.

[19] Q.-Y. Zhou and U. Neumann. 2.5d building modeling by discovering global regularities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 326–333. IEEE, 2012.

# APPENDIX

In this work, to train point classifier, a codebook containing 33 types of roof point semantics is created. Examples of all point semantics used in this work are shown in Figure 13. We then derive variants of the codes using base model shown in Figure 4 in two aspects. We then create hundreds of thousands of synthetic roof models to mimic real roof data in both various dimensions and degradations of erosion.

## A. MODEL SIZE DERIVATION

To generate models in various dimensions, when the models are created we make adjustments to a few parameters of base models such as height, slope of roof surface and the location of ridge and valley. Since it is inaccurate and unreasonable to assume any distribution for these parameters and we want to generate models that can cover as much as possible cases in real world, we evenly draw these parameters from their given ranges. We present the size derivation of each base roof in below.

1) **Flat**
A small slope is allowed for flat roof, so we change the angle between flat surface and ground plane from 0 degrees to 10 degrees.

2) **Gable**
We fix the width of gable roof and set it to be 10 meters. There are two parameters that are adjustable. First, the median ridge is shifted along the width side, which starts from the middle of the width and is shifted by 2.5 meters. Second, we set the slope of gable surface to be one between 20 to 75 degree.

3) **Hip**
In addition to how we configure the gable roof, the slope of two wing faces is changed from 20 to 75 degree.

4) **L-Joint**
For each branche, the same configuration of the gable roof is applied.

5) **T-Joint**
For each of two branches(vertical and horizontal), the same configuration of the gable roof is applied.

6) **X-Joint**
For each of two branches that pass the joint, the same derivation of the gable roof is applied.

7) **Gambrel**
Given a gambrel roof with two slope parts, the angle between each part of gambrel roof and Z direction is changed. Denoting the angle between the upper part of gambrel with Z is $a_0$ and the angle between the lower part with Z is $a_1$. $a_0$ and $a_1$ are changed from 20 to 75 degrees, which is subject to $a_0 - a_1 \geq$ 20 degree.

8) **Hex**
We synchronously change the slope of all side faces from 20 to 75 degrees.

9) **Mansard**
The same configuration of flat roof is applied to derive the top flat face of mansard. We then synchronously adjust the slope of two side faces from 20 to 75 degree.

10) **Pyramid**
The same configuration of hex roof is applied here.

11) **Shed**
We change slope from 20 to 75 degree.

12) **Loft**
For the gable branch of loft roof, the same configuration of gable roof is applied. For the slope surface that gable is connected to, the slope of this surface is changed from 20 to 75 degree.

## B. MODEL EROSION DERIVATION

To mimic erosion shapes of real models, we refer to a method proposed in [8], and three filter scales $\alpha_0 = 3, \alpha_1 = 6, \alpha_2 = 9$ meters are employed to generate eroded roofs under three scales.

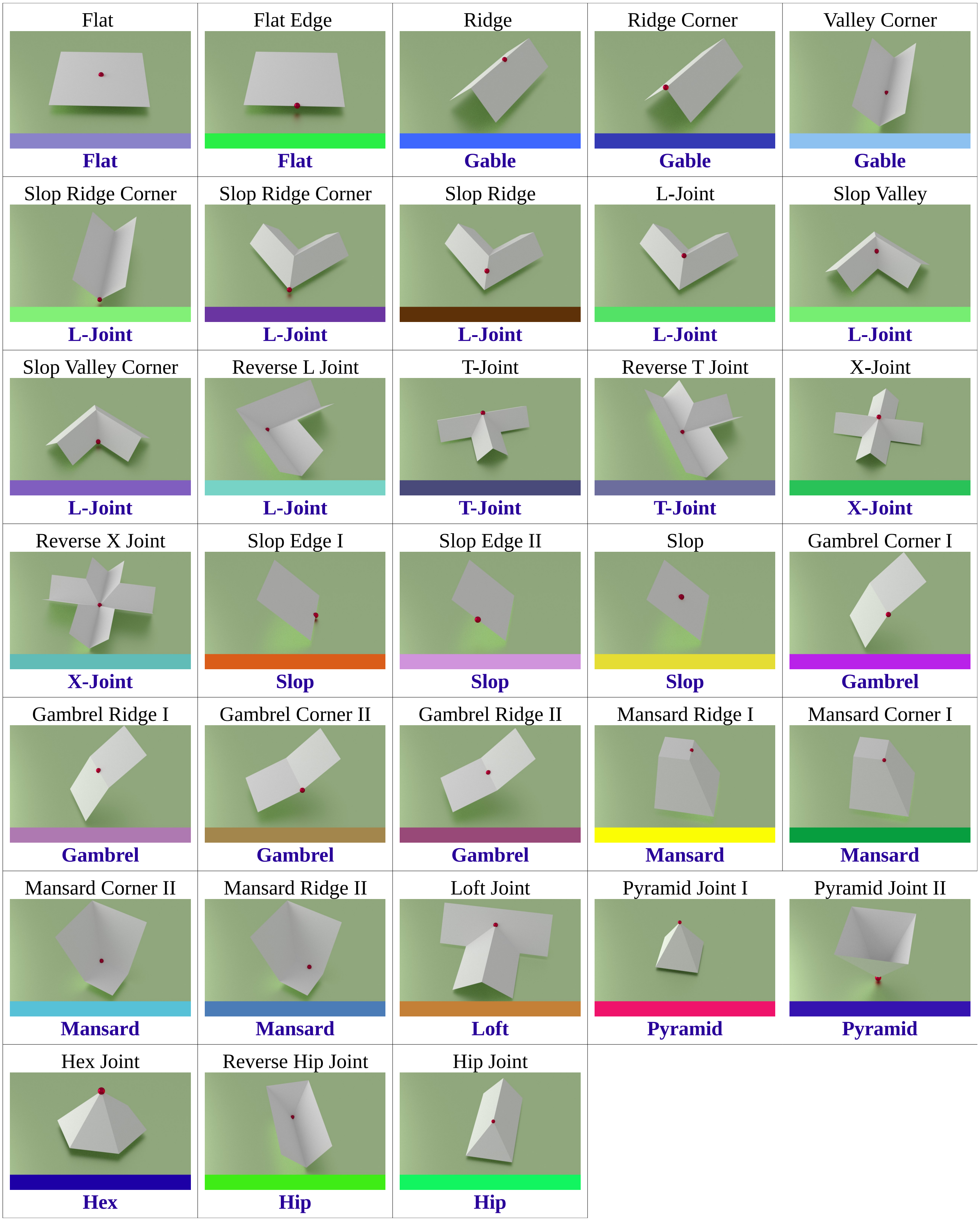


**Figure 13: Illustration of the key points (marked in red) used in the proposed approach. The color bar under each image corresponds to all color used to render corresponding roof points with the same semantics in this paper. The name of the base model of each sample is highlighted in blue under each picture.**