Foliage Simplification Based on Multi-viewpoints for Efficient Rendering

Sulan Zhang[a]
[a] The College of Mathematics and Computer, Yangtze Normal University, Chongqing 408100, China
Email: slzhang@cqu.edu.cn

***Abstract*— Vegetation is an essential component of the outdoor scene in the fields of virtual environment and computer game. The foliage models consisting of a great number of complex structures make real-time visualization impossible. In this paper, a novel viewpoint-driven foliage simplification framework is proposed for efficiently rendering virtual plants in polygonal models. Before the collapsing step, complex leaf models are reshaped as quadrilaterals and then separated into clouds of cells for rapidly finding the best leaf pair. Unlike the geometric foliage simplification approaches, we introduce an information-theoretic tool, mutual information to measure the leaf visibility and the leaf-collapse error from multi-viewpoints. The viewpoint-dependent foliage simplification algorithm produces foliage level of detail (LoD) models close to the original ones in terms of visual and geometric criteria. Our approach is appropriate for applications which require exact geometry tolerance but also high visual quality.**

***Index Terms*— real-time visualization, mutual information, leaf-collapse, LoD, visual quality.**

I. Introduction

VEGETATION is essential in virtual environment or computer game [1], [2]. There have been many successful systems for plant architecture modeling and growth simulation such as AMAP [3], L-system [4], Xfrog [5]. However, all these models are formed by such a great number of complex plant structures that real-time visualization is not possible. The case is worse for plant community with massive amount of data in detailed foliage, independent, small in size, repetitive in distribution, which paralyzes the most advance available rendering system. Generally, the highly detailed models are not always necessary to be rendered at full scale especially when the viewers are far away from the objects. Multi-resolution level of detail (LoD) models are well-know methods in the literature for reducing the polygonal complexity of models to improve performance in rendering highly detailed meshes [6].

The basic scheme for constructing LoD approximation of models is simplification. Unlike the continuous surfaces [7], the foliage composed of sparse non-connected geometry cannot be optimally simplified with common methods as edge collapses [8]. According to the nature of the foliage, some special simplification schemes are designed to construct the multi-resolutions models such as those based on leaf-collapse [9] or others based on pruning [10]. Most polygonal foliage simplification methods adopt leaf-collapse simplification scheme and the geometric similarity as a measure of the quality between an original mesh and the one obtained from simplification. With these methods we can achieve foliage models that are very geometrically similar to the original models in low temporal cost, however the visual quality is neglected. The appearance of the simplified models may be seriously distorted especially when they are in very coarse level. In many works, information entropy has been studied to measure the correlation between a set of viewpoints and visibility of the objects [11], [12]. They suggest that the variation on entropy has a close relationship with the model silhouette during the simplification process.

The purpose of this paper is to propose a foliage simplification framework that can keep as much visual quality as possible for efficient rendering. Before implementation of our foliage simplification algorithm, the complex leaf mesh is transformed into a quadrilateral so that all species of leaves in the foliage, either broad or thin are represented with the uniform form. In our foliage simplification algorithm, we introduce a new viewpoint-driven simplification algorithm based on an information-theoretic measure called foliage mutual information. This metric measures the correlation between a set of viewpoints and the leaves of the foliage. We then develop the leaf visibility as well as simplification error for the foliage simplification. The leaf visibility quantifies how much visual information of a single leaf possesses while the error metric assesses the variation of the foliage mutual information as the cost of a leaf-collapse operation.

The rest of the paper is organized as follows. We survey the previous foliage simplification approaches for efficient rendering in section II. Next, we discuss the complex leaf mesh simplification in our framework before implementing the foliage simplification algorithm. In section IV, we define the leaf visibility and the simplification error metric based on mutual information to find the best leaf pairs as well as measure the leaf collapse cost. In section V, we describe the viewpoint-driven foliage simplification algorithm. In section VI and VII, we introduce the implementation of experiment and explicit the simplified results of our simplification algorithm comparing the pure geometric simplification algorithm. Finally, conclusions

Manuscript received February 4, 2013; revised July 8, 2013; accepted November 22, 2013.
This work has been supported by Natural Science Foundation Projects of CQ (CSTC 2010BB2213, CSTC 2011BB2073) and Fundamental Research Funds for the Central Universities (CDJRC10180010).

doi:10.4304/jsw.9.7.1655-1665

and future work are presented.

II. Related Works

Researches on the real time visualization of detailed plants aim at decimating the primitives of the plants to a certain level so that the large-scale scenes could be efficiently rendered without overwhelming the current graphics hardware. Rendering plants in real-time has been extensively studied, and many methods have been developed. Depending on the presentation of the virtual plants, the previous works can be broadly classified into three representative categories: image-based algorithms, point-based algorithms, and polygon-based algorithms.

Image-based rendering: In the field of image-based rendering, imposters and billboards [13] are two common methods used to realize efficient rendering of the plant models especially for trees covered with dense leaves, however they suffer parallax artifacts at medium-close distances due to weak geometry. Since then, Max [14], [15] adds depth information to the pre-calculated images for imposters. Later on, Shade et al. [16] and Chang et al. [17] introduce layered depth image (LDI) to render objects from pre-computed pixel-based representations with depth from different viewpoints. To solve the parallax problem, Garcia et al. [18], [19] use textures to increase the detail of the leaves without increasing the memory, and Jakulin [20] presents trees with sets of parallel billboards to eliminate the artificial effect. Recently, some researchers have developed the billboard clouds [21], [22] to represent a plant by a set of arbitrarily oriented billboards. Others work with volumetric textures like Meyer et al. [23] converting complex natural object into volumetric textures. Although those improvements make better rendering at the medium-close distances, they are still not suitable for real-time rendering at close distance.

Point/Line-based rendering: Point/line models [24] are efficient for rendering with small polygons, and usually combine with the polygon to construct hybrid models for the trees. At close distances, polygonal geometry is used. With increasing viewing distance, branch meshes would be transformed into lines and leaves would be instead of points. For even farther distance, fewer points and lines will be used by randomly sampling [25] or by merging small points into one [26]. Point-based rendering of trees is only efficient for distant objects since the effect is visually unacceptable closely.

Polygon-based rendering: Polygon is the primitive to present the geometric model of an object. Many polygon based simplification methods have been explored to eliminate some geometric details for speed. These methods are efficient in simplifying objects with continuous surfaces, but not available to the foliage consisting of many isolated surfaces. So some special approaches have been proposed for the foliage simplification.

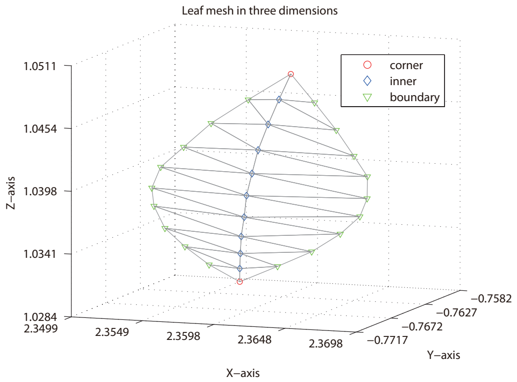
The Foliage Simplification Algorithm (FSA) [9], [27] is based on leaf-collapse operation in which a pair of leaves are recursively selected and replaced by a new lager leaf with similar geometry. In order to find the best leaf pair to be collapsed, the cost function based on geometric similarity for leaf-collapse is extended to area, diameter and so on in Progressive Leaves Union (PLU) [28]. Hierarchical Union of Organs (HUO) [29] improves PLU, in which the presentations of leaves in the collapse operation are extended to triangular leaves and the hierarchy is introduced to simplification. Later, the idea of hierarchy is extended to simplify leaves not only phyllotaxy, but also the topology of branches [30]. The methods mentioned above can simplify broad leaves well, but not coniferous leaves, Deng et al. [31] proposed a method for this foliage by using two representations, cylinder and line, to represent close and far coniferous leaves respectively. Later on, Deng et al. [13] introduce a GPU-oriented design of LoD storage structure and uneven subdivision of the tree crown volume for decreasing the LoD model construction cost and the communication between CPU and GPU in rendering.

All these polygonal foliage simplification methods share a similar basis: to decimate geometry by recursively combining two leaves into one, which also known as the leaf-collapse operation. In the previous foliage simplification methods, the error metric for the leaf-collapse operation is based on geometric similarity. Those geometric approaches concern on geometric fidelity while neglect visual quality, they produce foliage approximations with a very low geometric error but poor visual quality especially when the foliage models are simplified to the very coarse levels. Here we address the problem from both points and develop a viewpoint-driven foliage simplification approach capable of taking both geometric effect and visual effect into consideration.

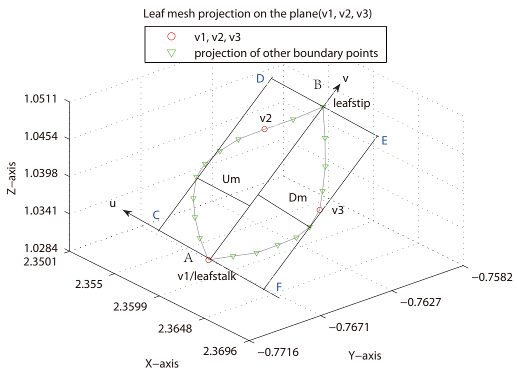
III. Complex Leaf Mesh Simplification

The broad leaves in nature may have various shapes rather than quadrilaterals. In order to make our collapsing algorithm available for different shapes of leaves like broad and thin, the leaf meshes of different species should be in same presentation. So before the foliage simplification algorithm, the complex leaf meshes are desirable to be simplified into quadrilaterals first. No matter how complex a leaf, the mesh is always composed of triangles and vertices which can be classified into three categories: inner, boundary and corner points. The corner points include the leafstalk and leaf tip points, which are nearest and furthest points from the branch bearing the leaf. Fig.1(a) gives an example of the complex leaf mesh for Corylus avellana.

The complex leaf mesh simplification aims at projecting the complex leaf model on a plane, and then generating a quadrilateral as similar to the original leaf geometry as possible. According to the leaf special structure and its close relationship with the bearing branch, the complex leaf mesh simplification is separated into three steps. First is to construct a projection plane $f(x, y, z) = ax + by + cz + d$ with three points v_1, v_2, v_3 which satisfy $f(v_i) = 0\{i = 1, 2, 3\}$. The leafstalk point is the joint of



(a) Complex leaf mesh



(b) Quadrilateral $CDEF$

Figure 1. Complex leaf mesh simplification. (a) Complex leaf mesh for Corylus avellana, the points of which are divided into three categories: corner points including leafstalk and leaf tip points marked as round, the boundary points marked as rhombus and the inner points marked as triangle. (b) Quadrilateral presented as $CDEF$ generated from complex leaf mesh simplification operation.

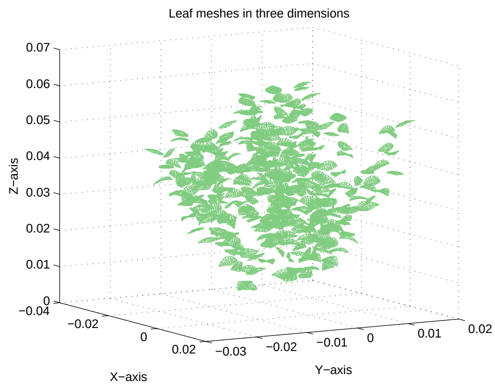
the leaf and the branch bearing the leaf, so it is necessary to keep the leafstalk point in the generated quadrilateral plane to preserve the realism. Therefore the first point v_1 is determined as the leafstalk point. Other two points v_2, v_3 are from the remaining points in the leaf mesh and satisfy the sum of the distances from all points in the leaf mesh to the plane f_{v_1,v_2,v_3} determined by v_1, v_2, v_3 is minimal. The minimal sum of distances is expressed as:

$$min\left\{\sum d(v_i, f_{v_1,v_2,v_3})\{v_i \in V - \{v_1, v_2, v_3\}\}\right\}, \quad (1)$$

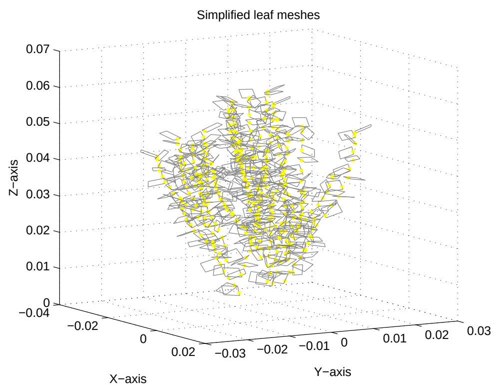
where v_1 is the leafstalk point, $v_2, v_3 \in \{V - v_1\}$, V is the set of total vertices of the leaf mesh, and d represents the function computing the distance from the point v_i to the plane determined by points v_1, v_2, v_3. With Eq. (1), the fittest three points v_1, v_2, v_3 for the projection plane are found in the leaf mesh and marked as round in Fig. 1(b).

Next, all the boundary points in the leaf mesh are projected onto the formerly generated plane. Fig.1(b) shows us the projection result of the leaf mesh in Fig.1(a) where round points determine the projection plan while the triangle points are the projection of other boundary points. Finally, with the projection plane $f(v_1, v_2, v_3)$ and the projected boundary points of the leaf mesh, the quadrilateral is constructed as follows. We take the leafstalk point A as the center of the 2D coordinate system on the projection plane and the line AB as x axis where point B is the leaf tip point projection in Fig.1(b). Then the projected boundary points are separated into two parts by x axis, the up points above the x axis, and the down points below the x axis respectively. Let U_m save the maximal distance of up point set to line segment AB and D_m save the maximal distance of down point set to line segment AB. Then we can map the leaf contour to a rectangle named $CDEF$ as Fig.1(b) where C, D, E, F denote the four coplanar points in projection plane. With the projection operation, a leaf could be drawn in a rectangle, so the complex leaf mesh is finally replaced by a quadrilateral which is the most close to the leaf mesh geometry.

Fig.2 presents the foliage original model of Corylus avellana and the result after the complex leaf mesh simplification operation in three dimensions. The two foliage models before and after simplification are a little different which means the quadrilateral model suffers some loss of validity. However, with the mesh simplification operation, the complex leaf mesh is simplified into a quadrilateral which has the same form with thin leaf mesh. So foliage with either broad or thin leaves could be simplified with our viewpoint-dependent simplification algorithm.



(a) Original model



(b) Results from leaf mesh simplification

Figure 2. Original Corylus avellana corresponding to its leaf mesh simplification result. (a) is the original foliage model of Corylus avellana, (b) is the simplified foliage model, the points in yellow mark the leaf stalk points.

IV. Viewpoint-dependent Simplification Metric

After the complex leaf mesh simplification, leaves are presented in quadrilaterals as the inputs of the foliage simplification algorithm. Before defining the foliage simplification algorithm to construct the LoD mode of the foliage for efficient rendering, the simplification metric should be determined. In previous works, the simplification metric to determine which pair of leaves should be firstly collapsed mostly focuses on geometry like distance, area, planarity similarities, while the visual quality of foliage from a set of viewpoints is neglected. In this section, we will introduce information-theoretic measure called mutual information into simplification metric, so that the appearance of the foliage could be well preserved when a large number of leaves are decimated during the foliage simplification algorithm.

A. Information-theoretic Measures

In order to measure the visual information of the foliage captured from a set of viewpoints, some information-theoretic tools like entropy and mutual information are utilized, the concepts of which are defined as follows.

Let A be a finite set, X be a random variable taking values x in A with distribution $p(x) = Pr[X = x]$, and let Y be a random variable taking values y in B with distribution $p(y) = Pr[Y = y]$. The *Shannon entropy* [32], [33] $H(X), H(Y)$ of random variables X and Y are defined by

$$H(X) = -\sum_{x\in A} p(x)\log p(x), \tag{2}$$

$$H(Y) = -\sum_{y\in B} p(y)\log p(y). \tag{3}$$

The Shannon entropy $H(X), H(Y)$, measure the average uncertainties of random variable X and Y. All logarithms are based 2 with the convention that $0\log 0 = 0$ is used for continuity. Generally the entropy is presented in bits.

The *conditional entropy* between two distributions about variables X, Y is defined as follow equation:

$$H(Y|X) = -\sum_{x\in A} p(x)\sum_{y\in B}\log p(y|x), \tag{4}$$

where $p(y|x) = Pr[Y = y|X = x]$ is the conditional probability. The conditional entropy $H(Y|X)$ measures the average uncertainty associated with Y if we know the outcome of X. In general, we have $H(Y|X) \neq H(X|Y)$ and $H(X) \geq H(X|Y) \geq 0$. Then, the *mutual information* (MI) between X and Y sets is defined by

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= \sum_{x\in A} p(x)\sum_{y\in B} p(y|x)\log(p(y|x)/p(y)). \end{aligned} \tag{5}$$

The mutual information $I(X,Y)$ is a measure of the shared information between variables X and Y. From Eq.(4), it can be inferred that $I(X,Y) = I(Y,X) \geq 0$.

One of the features we associate with goodness or quality of a viewpoint is the amount of information it provides us with. We assume that the information that the viewpoints provide is visibility. Recently, Information theory tools have been used to select good viewpoint for a object [12]. Here we define viewpoint mutual information that allows us to obtain the leaf visibility from a set of viewpoints. The viewpoint mutual information incorporates both the projected area and the number of leaves, and can be understood as the amount of information captured from the viewpoints.

B. Foliage Mutual Information and Leaf Visibility

To compute the amount of visibility about leaf clouds L' in the foliage from a set of viewpoints V', we define an information channel $V \rightarrow L$ between the random variables V and L, which represent the set of viewpoints and the set of leaves respectively. Viewpoint will be indexed by v and leave by l. The marginal probability distribution of V is given by $p(v) = 1/N_v$, where N_v is the number of viewpoints. Here we suppose that the probability of each viewpoint is the same. The conditional probabilities $p(l|v)$ are given by the relative area of the projected triangles over the sphere of directions centered at viewpoint v and can be written as $p(l|v) = a_i/a_t$, where a_i is the area of the current leaf i projected over the sphere, and $a_t = \sum_{i=0}^{N_l} a_i$ is the total area of the whole foliage sphere. N_l is the number of leaves in the foliage, a_0 represents the projected area of the background in open scenes and is assigned to 0 when the background is not taken into consideration. Finally, The marginal probability distribution of L is given by $p(l) = \sum_{v\in V'} p(v)p(l|v) = (1/N_v)\sum_{v\in V'} p(l|v)$.

From this channel, it follows that the conditional entropy (4) can be written as

$$\begin{aligned} H(L|V) &= -\sum_{v\in V'} p(v)\sum_{l\in L'} p(l|v)\log p(l|v) \\ &= (1/N_v)\sum_{v\in V'} H(v), \end{aligned} \tag{6}$$

where $H(v) = -\sum_{l\in L'} p(l|v)\log p(l|v)$ is the entropy of view point v. With the conditional entropy, The mutual information (5) is given by

$$\begin{aligned} I(V,L) &= \sum_{v\in V'} p(v)\sum_{l\in L'} p(l|v)\log(p(l|v)/p(l)) \\ &= (1/N_v)\sum_{v\in V'}\sum_{l\in L'} p(l|v)\log(p(l|v)/p(l)) \\ &= \sum_{l\in L'}(1/N_v)\sum_{v\in V'} p(l|v)\log(p(l|v)/p(l)) \\ &= \sum_{l\in L'} I(V,l), \end{aligned} \tag{7}$$

where

$$I(V,l) = (1/N_v)\sum_{v\in V'} p(l|v)\log(p(l|v)/p(l)). \tag{8}$$

The foliage mutual information expresses the degree of dependence or correlation between a set of viewpoints and the foliage, and tells the amount of visual information for the whole foliage. It can be used as a tool for weighting foliage simplification result that is wether the simplification method cloud well preserve the visual quality of the original foliage. For a plant, the value of foliage mutual information always depends on the number of viewpoints and their distribution around the foliage. In the work of Lindstrom et al [34], they suggest that using uniformly distributed view points around the foliage could obtain good results. Moreover, they found when the number of viewpoints exceeds 20, it adds little accurate information for the mesh simplification.

For a single leaf, we propose to take $I(V, l)$ described in Eq.(8) as its visual quantity measurement called leaf visibility from a set of viewpoints V' which are uniformly distributed around the foliage. High value of leaf visibility means the high importance of the leaf for the whole foliage appearance, while low value corresponds to low importance. Collapsing the leaf with low value of visibility has little effect on the contour of the foliage while the geometry is gradually reduced. The leaf visibility provides a measure for deciding which leaf should be firstly merged into a new one at the leaf collapse step.

C. Viewpoint-dependent Foliage Simplification Error Metric

In the foliage simplification algorithm based on multi-viewpoints, the leaf l with lowest visibility should be firstly united with the one of its geometrical neighbors l' to keep the visual as well as geometrical quality of the simplified foliage. In [35] mutual information has been used to describe the shape for object recognition since it's sensitive to the shape variation. In addition, the mutual information evaluates the average variation from a set of viewpoints covering the sphere of foliage so that changing the orientation of the foliage no long leads to a different simplification result.

Taking into these facts into account, our simplification error metric (C_p) for collapsing a pair of leaves $p(l, l')$ is defined by foliage mutual information variation for all viewpoints:

$$C_p = |I(V, L) - I(V, L')| \,, \tag{9}$$

where L' represents the simplified foliage. The simplification error metric indicates the variation before and after performing the leaf-collapses. The smaller value of C_p is, the less loss of visual quality for the simplified foliage will be. In order to avoid quadratic number of comparisons when looking for the leaf pair with lowest visual error, we cluster the leaves into a cloud of cells with Octree partition [36]. With this structure, when looking for leaf pair for collapsing in one cell, its neighboring cells don't need to be considered. Our foliage simplification error metric based on mutual information is sensitive to the distance of viewpoint to the foliage. Therefore an equal radius of the viewpoint sphere for the foliage is adopted necessarily.

With regard to computation of the leaf visibility and mutual information for the whole foliage, several techniques based on projected areas have been analyzed in [37]. They are the OpenGL histogram, the hybrid SW-HW histogram and the occlusion query. In this paper we use hardware occlusion queries to calculate the number of pixels that pass the z-buffer test as the projected area for each leaf. Basically, for each viewpoint, the foliage is rendered twice to obtain the number of non-occluded pixels visible from the current viewpoint. First, the foliage is sent for rendering and the depth buffer is initialized. Second, we independently sent each leaf for rendering. With this procedure it's necessary to make $N_l + 1$ rendering passes, N_l being the number of leaves in a plant. Only in the first pass the whole geometry is rendered, while in the following passes, one single leaf is rendered again to count the number of pixels that pass the depth test and thus the number of pixels which are visible from the current viewpoint.

V. Viewpoint-driven Foliage Simplification Algorithm

The foliage simplification approach proposed in our paper, like many previous foliage simplification algorithms, is based on leaf collapsing technique. The main idea of leaf collapse operation is that the new leaf generated from collapsing the current best leaf pair maintains the most geometric similarity as the original leaf pair. In the former section, we have introduced the foliage mutual information and leaf visibility to measure the visual information of the whole foliage and a single leaf from a set of viewpoints around them. In our leaf collapse step, the leaf l_1 with least visibility is optimal to be chosen as one of the best leaf pair currently. Another candidate leaf l_2 of the best leaf pair is from the same cell generated from Octree method and incurs least simplification error C_p defined as Eq.(9). So the simplified foliage preserves the visual effect and the geometric effect of the foliage well when the number of leaves is gradually reduced.

A. Leaf Collapse Operation

In first step, we have simplified the complex leaf mesh to a quadrilateral, now all the leaf meshes in the foliage are in uniform presentation with four verities and two adjacent triangles. In the leaf collapse step, a new leaf is generated to approximately represent the space two original leaves have occupied. The vertices of the new leaf will be from the two collapsed leaves, and no new vertices are introduced. This method will allow us to maintain an area similar to the two original leaves, so the geometric validity of the foliage during the leaf collapse process could be well kept. The following describes the leaf pair collapsing process.

For fusing the two candidate leaves into a new one, the two vertices v_1, v_2 with the longest distance are firstly found from the eight vertices of the two original leaves. Those two vertices would be the first two vertices and the longest distance would be the diameter of the new leaf. Other two vertices of the new leaf v_3, v_4 are chosen from

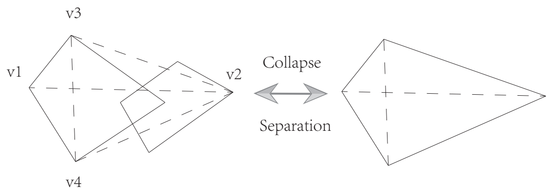


Figure 3. The leaf collapsing process.

the remaining vertices of the two original leaves one by one satisfying the sum of distances to the chosen vertices in former step is maximum. Then the normal vectors for the four vertices of the new leaf are computed and a new quadrilateral is generated from the four vertices as the new leaf. The merging process is shown as Fig.3.

The leaf collapse method mentioned above is based on area maximum, so the new leaf keeps as much geometric similarity as possible during the collapsing process. The leaf collapse metric is used in our foliage simplification algorithm, then the geometric validity of the foliage would well kept during the simplification process. Next we will introduce our algorithm of foliage simplification.

B. Foliage Simplification Algorithm Based on Multi-viewpoints

Leaf visibility provides a quantitative measure for the amount of visual information which a certain leaf contributes to the appearance of the whole foliage from a set of viewpoints. From the definition as Eq.(8) in former section, it's deduced that the more visual information a leaf possesses, the more important role it plays for keeping the silhouette of the whole foliage. Whereas, the leaf with lowest value of visibility is most insignificant than others in the visual effect of the whole foliage, merging it firstly with its geometric neighbors would introduce least appearance loss. In each leaf collapse step of the foliage simplification, the leaf with insignificant visibility in the remaining foliage is chosen as one of the pair of leaves to be collapsed next. So in our simplification process, the visibility for all the initial leaves in the foliage is computed firstly and stored for accessing how valuable a leaf is in the whole foliage visual quality.

In order to keep the visual and geometric qualities of the foliage, the pair of leaves to be collapsed should be both geometrical similar and incur least visual error. The leaf with least visibility in the current foliage is fixed on as one of best leaf pair l_1. The other candidate leaf l_2 composing the best leaf pair is from the geometrical neighbors of leaf l_1 which are in the same cell generated from space segmentation with Octree. If the number of leaves in the current cell where leaf l_1 locates $N(C)$ is less than a pre-determined const N', we extend the cell to its geometric neighbor cells in 3D space. Appropriate value of N' is important for searching leaf collapse pair. If N' is too large, more time will be taken. On the other hand if N' is too small, more suitable leaf pair may be lost. When assigning the value of N', a tradeoff between accuracy and cost should be considered.

When the appropriate pair of leaves is found, leaf collapse operation described as Fig.3 is performed. After the leaf collapse, the leaf pair $p(l_1, l_2)$ is remove from current foliage while the newly united leaf l is added into the current mesh model. A leaf collapse in our algorithm may affect the visibility of the remaining leaves in principle. Because the new leaf l is similar to the two original pair of leaves $p(l_1, l_2)$, but they not just equal, so when a new leaf replaces the pair of collapsed leaves, some valid leaves in foliage may be sheltered while other leaves in shade may come out. But this does not happen to every leaf in the foliage. At each step we choose only a small group of leaves that are affected by a leaf collapse, then the visibility is recalculated for those leaves. The following pseudo code shows the summary of our simplification algorithm.

Algorithm 1 Foliage simplification algorithm based on multi-viewpoints

// Compute initial visibility for each leaf l in foliage mesh L, and build a queue q for visibility value.
for $(l \in L)$
  Compute $I(v, l)$, where $v = 1, \cdots, n$
  Insert double (l, I_l) in queue q
end for
// Perform leaf collapse operation on the leaf with minimal visibility where M is the total number of leaf in foliage, and $N(C)$ represents the number of leaves in cell C.
$m = M$
while $(m > 1)$
  Choose leaf l_1 with minimal visibility I in queue q
  Find cell C which leaf l_1 belongs to
  while $(N(C) < N')$
    Enlarge cell C with surrounding cells
  end while
  // Find the pair of leaves $p(l_1, l_2)$ with lowest C_p in cell C. L, L' are the foliage before and after a leaf collapse operation.
  $C_{Pmin} = MAXIMUM$
  for $(l_i \in \{C - l_1\})$
    Collapse the pair of leaves $p(l_1, l_i)$
    Compute collapse cost $C_p = |I(V, L) - I(V, L')|$
    if $(C_p < C_{Pmin})$
      $C_{Pmin} = C_p$, $l_2 = l_i$
    end if
    Undo the leaf collapse with leaf pair $p(l_1, l_i)$
  end for
  // Perform the leaf collapse operation.
  Find leaf pair $p(l_1, l_2)$ with lowest $C_{p_{min}}$ in q
  Collapse leaf pair $p(l_1, l_2)$ and generate a new leaf l'
  Delete l_1, l_2 and add the new leaf l' in queue q
  Recalculate the visibility of every leaf in the cell of leaf pair $p(l_1, l_2)$ and update their value in queue q
  $m = m - 1$
end while

In general, multi-resolution representations of foliage model are constructed from the viewpoint-dependent foliage simplification algorithm. The original mesh of the foliage is named as F_0, and other different approximations compose a serial set recorded as $F_1, F_2, \cdots, F_{n-1}$ where n is the times of leaf collapse operation. The data is organized as a binary tree, where the root-node is the leave that forms F_{n-1}, and the leaf-nodes are the leaves of the original foliage model F_0. With those multi-resolution representations, we can easily get any LoD foliage for real-time visualization.

VI. Implementation

In the programming model, several classes are defined. As leaf is the basic element in the simplification including a great number of faces and vertices, it consists of a list of vertices, a list of faces noted by the indexes of vertices, and its interrelated attributes such as area, diagonal length, normal, visibility, union age and so on. The methods in leaf are computing leaf area, diagonal, and normalizing the orientation of leaf polygon, et. al. To avoid collapsing leaf pair in the whole space, the foliage is firstly divided into clouds of cells with Octree [36] method. The cell is marked with indicating indexes *i*, *j*, a list of leaves, as well as its attributes including volume, number of leaves in the cell, and the field *E* to enable or disable the cell. The interfaces in cell are designed for computing the cell volume, adding or deleting leaf in the cell, collapsing leaf pair to reduce the cell geometry. The members in the foliage class are list of cells, number of cells, the active leaves remaining in the foliage, viewpoints, as well as the hardware occlusion query object. More details for the classes and their relationship are shown as Fig.4.

At the beginning of our experiment, the ARB occlusion query language of OpenGL is employed to collect the mutual information for the foliage and leaf visibility. Firstly, the *CreateArbQuery*() function of class defined as Fig.4 is called to create a querying body, *glQuery*, and the whole foliage is rendered in a certain viewpoint to initial the depth buffer. Then, a single leaf is rendered again to compute its projecting area on the sphere with function *RendertoQuery*(). At last, with the help of equations defined as Eq.7 and Eq.8, the *ComputeViewInformation*() is invoked to obtain the foliage mutual information and the single leaf visibility. Due to the limitation of the observing scope, only part of leaves could be captured from a single viewpoint, therefore the 20 viewpoints distributed over the vertices of regular dodecahedron are set in our algorithm. Fig.5 shows the trees under different viewpoints.

Before the simplification step, the function *LeafPolygonSim*() is called to convert the polygonal model of leaf to quadrilateral, so that all the leaves in the foliage have an uninform presentation for leaf collapsing operation. In addition, proper cells are generated from *CellUnion*() or *CellPartition*() operation, then the simplification is implemented on the respective cells with *Simplification*() operation. The simplification algorithm is performed on the hardware platforms of Intel Core i5 2.8GHZ with 4GB RAM and AMD Radeon HD 6450 1GB graphics card.

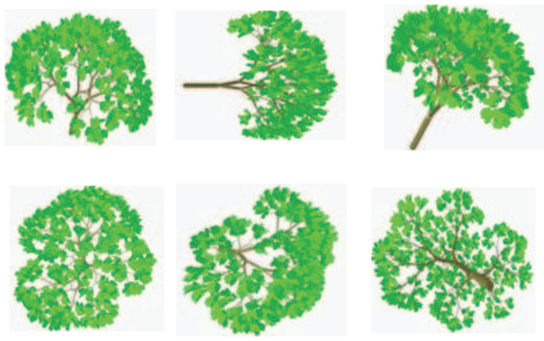


Figure 5. The images for tree rendering under different viewpoints.

VII. Result

We carried out tests with several plant models of differing complexities. To qualitatively measure the visual and geometric errors between the original and the simplified models, we implemented the root mean square error (RMSE) of the pixel-to-pixel image differences defined in [34] and the mesh comparison tool called Metro v4.07 [38] to compute the geometric differences. The generated results were compared with the those from geometric algorithms [13] at the same reduction level. In our simplification algorithm, three kinds of trees which are Corylus avellana, Purple willow and Temple juniper with broad, thin and needle leaf shapes were tested. Their geometry and the associated parameters in the simplification step are listed in Table I. The number of leaves and triangles in those three foliage models is increasing dramatically. Here we employed 20 viewpoints to collect the visual information and the distances from viewers or cameras to the centers of trees were predetermined as Table I to exactly capture the whole appearance of the trees. The original meshes of the former three tree models were are as Fig.6.

TABLE I.
THE GEOMETRY OF THE FOLIAGE MODELS AND PARAMETERS USED IN OUR TESTS.

Model	Geometry		Parameters for simplification	
	Leaves	Triangles	Viewpoints	Distance
Corylus avellana	386	772	20	2.8
Purple willow	4,456	8,912	20	3.0
Temple juniper	13,162	2,6324	20	3.2

In the following tests, the foliage models were reduced to 80, 60, 40, 20, 10, 5, and 2.5 percentage of original complexities with the algorithm based on mutual information and leaf visibility proposed in this paper, and the algorithm based on geometric similarity proposed in [13] separately. The detail geometry of the simplified foliage models is shown in Table II. With the increasing simplification ratio, the number of leaves and triangles in the foliage model decreases dramatically. For example, when the number of leaves in the foliage model is

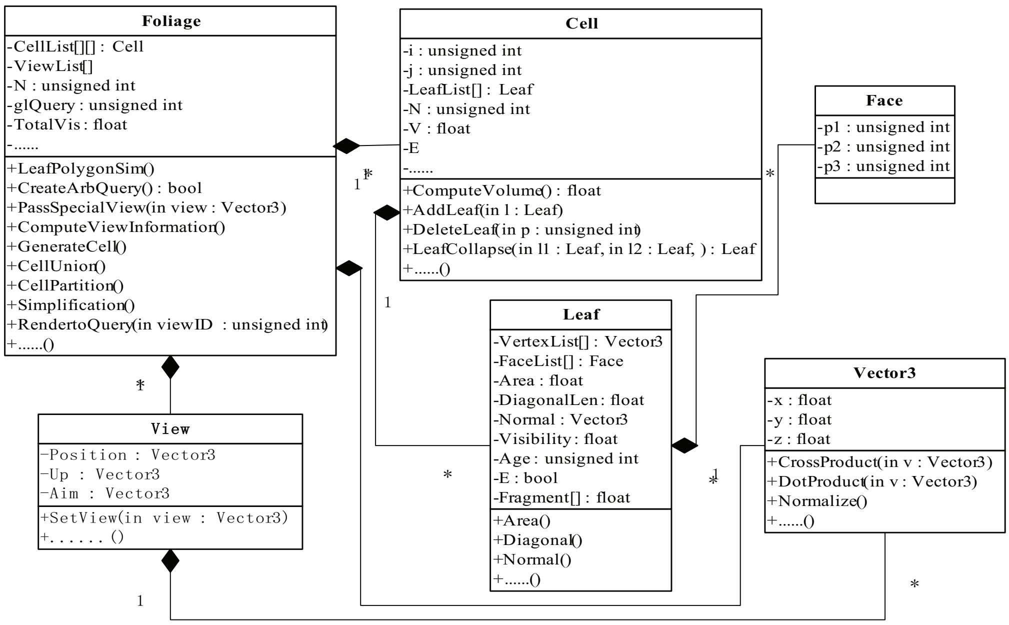
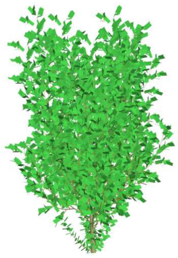


Figure 4. The definitions of classes and their dependencies in the foliage simplification implementation, and the number on the line as well as symbol '*' labels the aggregation between the classes.

(a) Corylus avellana



(b) Purple willow

(c) Temple juniper

Figure 6. Three kinds of tree models used for the simplification operation. (a) is Corylus avellana with 386 leaves, (b) is Purple willow with 4456 leaves and (c) is Temple juniper with 13162 leaves. Their geometry and parameters used in the simplification process are listed in Table I.

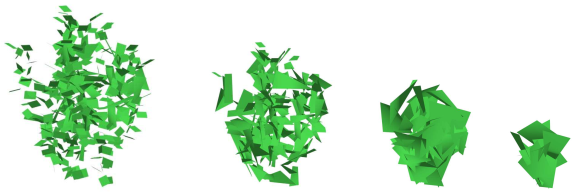
TABLE II.
THE GEOMETRY LEFT IN THE SIMPLIFIED FOLIAGE MODELS WHEN THE COMPLEXITY IS REDUCED TO 80, 60, 40, 20, 10, 5 AND 2.5 PERCENTAGE.

Reduction ratio	Corylus avellana		Purple willow		Temple juniper	
	Leaves	Triangles	Leaves	Triangles	Leaves	Triangles
80%	308	616	3,564	7,128	21,058	10,529
60%	462	231	2,673	5,346	15,794	7,897
40%	308	154	3,564	1,782	10,528	5,264
20%	154	77	1,782	891	5,264	2,632
10%	76	38	890	445	2,632	1,316
5%	38	19	444	222	1,316	658
2.5%	18	9	222	111	658	329

simplified into 2.5 percentage, there are only 9 leaves left in Corylus avellana, 111 leaves left in Purple willow and 329 leaves left in Temple juniper. The simplified results with the former methods for foliage models of Corylus avellana are as Fig.7, the results for Purple willow shown as Fig.8 and the results for Temple juniper shown as Fig.9.

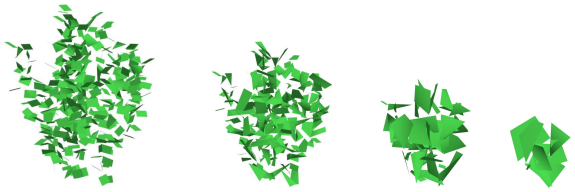
In our algorithm, we emphasize that the appearance preservation of the foliage when the simplification operation is carried out step by step. The leaf with least leaf visibility defined as Eq.(8) is prior to collapsed. Meanwhile another candidate leaf to be collapsed is elected from the neighborhood and the leaf-collapse operation satisfying the visual loss of the whole foliage defined as simplification error metric in Eq.(9) is minimal. From results generated from the simplification, it's apparent that our algorithm maintains the appearance of the simplified foliage better even when the reduction ratio becomes very small. On the opposite, when the reduction ratio becomes smaller and smaller, the simplified foliage with geometric similarity method could not keep original distribution of the leaves, for example leaf density becomes much sparser. When the LoD is reduced to a certain sparse level, the simplified result even loses the original appearance. Therefore, our simplification algorithm keeps foliage appearance better than the algorithms which only focus on the geometric similarity.

Fig.10 drawn from the errors on RMSE and Hausdorff distances clearly depicts the differences between



80% 40% 10% 2.5%

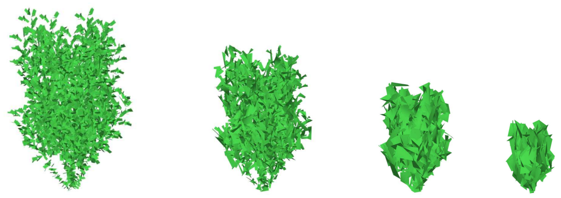
(a) Foliage models of Corylus avellana simplified with our algorithm



80% 40% 10% 2.5%

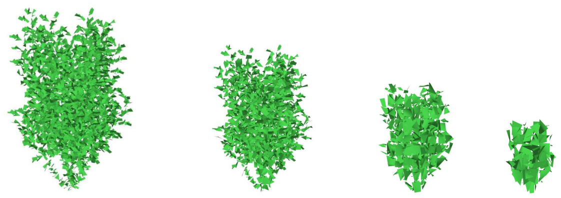
(b) Foliage models of Corylus avellana simplified with algorithm proposed by Deng [13]

Figure 7. The simplified results for foliage model of Corylus avellana when the complexity of the original foliage model is reduced to 80, 40, 10, 2.5 percentage.



80% 40% 10% 2.5%

(a) Foliage models of Purple willow simplified with our algorithm

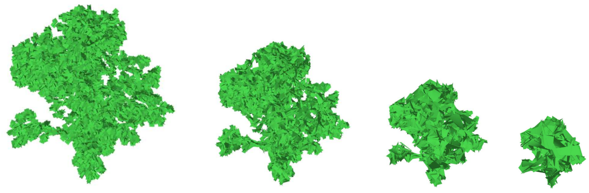


80% 40% 10% 2.5%

(b) Foliage models of Purple willow simplified with algorithm proposed by Deng [13]

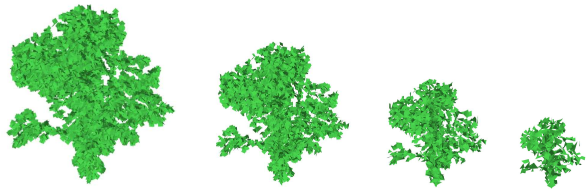
Figure 8. The simplified results for foliage model of Corylus avellana when the complexity of the original foliage model is reduced to 80, 40, 10, 2.5 percentage.

our algorithm and the geometric simplification approach proposed in [13] in visual and geometric aspects. The solid curves are the results of foliage simplification based on information entropy metric proposed in our paper while the dotted curves are the foliage simplification results obtained from the foliage simplification algorithm based on geometric similarity. In Fig.10(a), the curves of RMSE from the simplification of the three foliage models with our algorithm are lower than the curves from the results with geometric algorithm. It reveals that our foliage simplification algorithm reserves much more visual information than the geometric foliage simplification algorithm. Although in the leaf collapse step, the fittest partner for the current candidate leaf with least visibility to compose the best leaf pair is from its geometric



80% 40% 10% 2.5%

(a) Foliage models of Temple juniper simplified with our algorithm



80% 40% 10% 2.5%

(b) Foliage models of Temple juniper simplified with algorithm proposed by Deng [13]

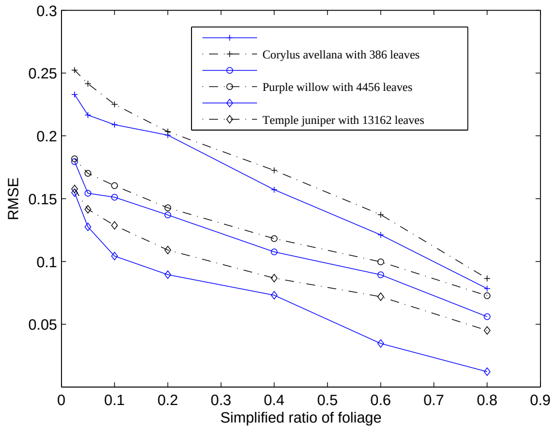
Figure 9. The simplified results for foliage model of Temple juniper when hen the complexity of the original foliage model is reduced to 80, 40, 10, 2.5 percentage.

neighborhood, the leaf collapse operation based on cost function defined as Eq.(9) may leads to some geometric loss shown as Fig.10(b) when the foliage simplification ratio decreases, especially in the foliage models with large number of leaves. However the loss is tolerable when the viewers are far from objects, that is our eyes concern more on the visual validity.

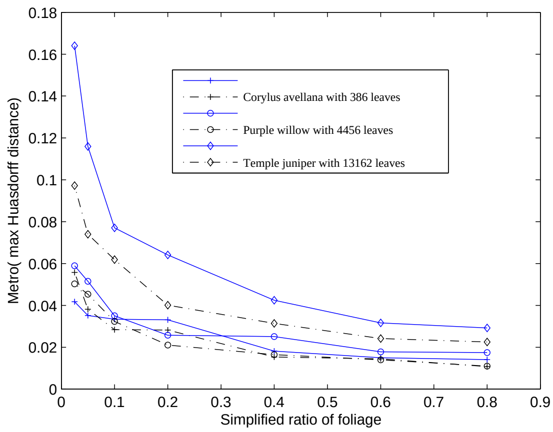
VIII. Conclusion and Future Work

This paper presents a viewpoint-driven foliage simplification method to extract any LoD model for foliage, and then large-scale vegetation scenes could be efficiently rendered on current hardware with limited capability where only a proper level of detail of the plants is rendered in the scenes. With leaf mesh simplification, the complex leaves are transformed into a uniform presentation, quadrilateral, therefore most species of foliage are available for the simplification algorithm. In the simplification algorithm, we introduce two information-theoretic measures called mutual information and leaf visibility from a set of viewpoints to weight the visual information of the whole foliage as well as the visual information of a certain leaf in the foliage. The leaf with lowest leaf visibility in the foliage is firstly collapsed to one of its geometric neighbors which induces lest variation in mutual information of the whole foliage defined as simplification error metric in our paper. As shown in the former experiments, our approach can maintain the foliage silhouette better than the geometric-based method and produce lower visual errors, mainly because it benefits from visibility information.

As future work, several potential further improvements may be explored. Firstly, as we know, many virtual plants modeling systems like L-system generate the topological structure of the plants according to which the leaves in the models could be grouped into ”botanically-faithful” unions. The best leaf pair for collapse is selected from



(a)



(b)

Figure 10. Curves of RMSE and Hausdorff distances between the original foliage models and the simplified foliage models. The solid curves are the simplified results with our algorithm while the dotted curves are obtained from simplified algorithm in [13].

those unions, and then the simplified results would be more faithful to the silhouette of the original foliage models than other space division methods. Secondly, the bottleneck in our algorithm resides in bus traffic, next step we may implement our algorithm on GUP, the data transferring between CPU and GPU could be avoided. At the same time we could take the advantage of the parallel computing capability of GPU to speed up the foliage simplifications algorithm. Thirdly, the leaf density may be introduced to the foliage simplification process. Generally, our eyes are insensitive to variation of the leaves in denseness, so leaf-collapse operation in the region with thickly leaf covering would incur less visual error than the sparse region.

ACKNOWLEDGMENT

This work has been supported by Natural Science Foundation Projects of CQ (CSTC 2010BB2213, 2011B-B2073) and Fundamental Research Funds for the Central Universities (CDJRC10180010).The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

REFERENCES

[1] S. Gregory, “Implications of a multidisciplinary students computer game design project,” *Journal of computer*, vol. 8, no. 2, pp. 1836–1840, 2012.

[2] Q. Xilong, L. Linfeng, and H. Zhongxiao, “Md3 model loading in game,” *Journal of computer*, vol. 7, no. 2, pp. 521–527, 2012.

[3] P. de Reffye, C. Edelin, J. Francon, J. M., and C. Puech, “Plant models faithful to botanical structure and development,” *ACM SIGGRAPH Comput. Graph.*, vol. 22, no. 4, pp. 151–158, 1988.

[4] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. Springer-Verlag, 1990.

[5] B. Lintermann and O. Deussen, “Interactive modeling of plants,” *Computer Graphics and Applications, IEEE*, vol. 19, no. 1, pp. 56–65, 1999.

[6] J. Ribelles, A. Lopez, O. Belmonte, I. Remolar, and M. Chover, “Multiresolution modeling of arbitrary polygonal surfaces: a characterization,” *Computers & Graphics*, vol. 26, no. 3, pp. 449–462, 2002.

[7] D. Xinyu and G. Guowei, “Elementary application of geometry face modelling under vrml complex object muster,” *Journal of computer*, vol. 6, no. 4, pp. 683–690, 2011.

[8] J. Cohen, M. Olano, and D. Manocha, “Appearance-preserving simplification,” in *SIGGRAPH, 1998 proceedings ACM*, 1998, pp. 115–122.

[9] I. Remolar, M. Chover, O. Belmonte, and J. Ribelles, “Geometric simplification of foliage,” in *Eurographics’02 Short Presentations*, 2002, pp. 397–404.

[10] J. Gumbau, M. Chover, I. Remolar, and C. Rebollo, “View-dependent pruning for real-time rendering of trees,” *Computers & Graphics*, vol. 35, no. 2, pp. 364–374, 2011.

[11] I. Viola, M. Feixas, M. Sbert, and M. E. Groller, “Importance-driven focus of attention,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 12, no. 5, pp. 933–940, 2006.

[12] M. Feixas, M. Sbert, and M. E. Groller, “A unified information-theoretic framework for viewpoint selection and mesh saliency,” *ACM Transactions on Applied Perception*, vol. 6, no. 1, pp. 1–25, 2009.

[13] Q. Deng, X. Zhang, G. Yang, and M. Jaeger, “Multiresolution foliage for forest rendering,” *Computer Animation and Virtual Worlds*, vol. 21, no. 1, pp. 1–23, 2010.

[14] N. Max, “Hierarchical rendering of trees from precomputed multi-layer z-buffers,” in *Eurographics Workshop on Rendering*, 1996, pp. 165–174.

[15] N. Max, O. Deussen, and B. Keating, “Image-based rendering using texture mapping hardware,” in *Proceeding of the 10th Eurographics Workshop on Rendering*, 1999, pp. 57–62.

[16] J. Shade, S. Gortler, L. He, and R. Szeliski, “Layered depth images,” in *SIGGRAPH, 1998 proceedings ACM*, 1998, pp. 231–242.

[17] C. Chang and A. Bishop, G. andLastra, “Ldi tree: a hierarchical representation for image-based rendering,” in *SIGGRAPH, 1998 proceedings ACM*, 1998, pp. 291–298.

[18] I. García and G. Patow, “inverse geometric textures,” *ACM Transactions on Graphics*, vol. 27, no. 5, pp. 1–9, 2008.

[19] I. García, M. Sbert, and L. Szirmay-Kalos, “Leaf cluster impostors for tree rendering with parallax,” in *Eurographics, 2005 short presentations*, 2005, pp. 69–72.

[20] A. Jakulin, “Interactive vegetation rendering with slicing and blending,” in *Eurographics, 2000 Conference-Short Presentations*, 2000, pp. 273–282.

[21] S. Behrendt, C. Colditz, O. Franzke, J. Kopf, and O. Deussen, “Realistic real-time rendering of landscapes using billboard clouds,” in *Proceeding of Eurographics*, 2005, pp. 507–516.
[22] A. Fuhrmann, E. Umlauf, and S. Mantler, “Extreme model simplification for forest rendering,” in *Eurographics Workshop on Natural Phenomena*, 2005, pp. 57–66.
[23] A. Meyer and F. Neyret, “Interactive volumetric textures,” in *Eurographics rendering workshop*, 1998, pp. 157–168.
[24] W. T. Reeves and R. Blau, “Approximate and probabilistic algorithms for shading and rendering structured particle systems,” *ACM SIGGRAPH Computer Graphics*, vol. 19, no. 3, pp. 313–322, 1985.
[25] M. Stamminger and G. Drettakis, “Perspective shadow maps,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 557–562, 2002.
[26] G. Gilet, A. Meyer, and F. Neyret, “Point-based rendering of trees,” in *Eurographics Workshop on Natural Phenomena*, 2005, pp. 67–72.
[27] I. Remolar, M. Chover, J. Ribelles, and O. Belmonte, “View-dependent multiresolution model for foliage,” *Journal of WSCG*, vol. 11, no. 1, pp. 370–378, 2003.
[28] X. Zhang and F. Blaise, “Progressive polygon foliage simplification,” in *International Symposium on Plant growth Modeling, simulation, visualization and their Applications*, 2003, pp. 217–230.
[29] X. Zhang, F. Blaise, and M. Jaeger, “Multiresolution plant models with complex organs,” in *International Conference on Virtual Reality Continuum and Its Applications*, 2006, pp. 331–334.
[30] Q. Deng, X. Zhang, and M. Jaeger, “View-dependent hierarchical foliage simplification,” in *Proceedings of the 2nd International Conference of E-Learning and Games*, 2007, pp. 33–44.
[31] Q. Deng, X. Zhang, S. Gay, and X. Lei, “Continuous lod model of coniferous foliage,” *International Journal of Virtual Reality*, vol. 6, no. 4, pp. 77–84, 2007.
[32] R. E. Blahut, *Principles and Practice of Information Theory*. Addison-Wesley, 1987.
[33] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Addison-Wesley, 1991.
[34] P. Lindstrom and G. Turk, “Image-driven simplification,” *ACM Transactions on Graphics*, vol. 19, no. 3, pp. 204–241, 2000.
[35] J. Rigau, M. Feixas, and S. M., “Shape complexity based on mutual information,” in *IEEE international conference on shape modeling and applications*, 2005, pp. 357–362.
[36] A. Gueziec, G. Taubin, F. Lazarus, and W. Horn, “Cutting and stiching: coverting sets of polygons to manifold surfaces.” *IEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, pp. 136–150, 2001.
[37] P. Castelló, S. M., C. M., and M. Feixas, “Techniques for computing viewpoint entropy of a 3d scene,” *Lecture Notes in Computer Science*, vol. 3992, no. 1, pp. 263–270, 2006.
[38] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: Measuring error on simplified surfaces,” *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 2001.

Sulan Zhang is a lecturer in the Department of Mathematics and Computer at the Yangtze Normal University. She earned her Doctor and M.S. degree in computer science from the University of Chongqing in 2009, 2013 respectively. She also earned her B.S. degree in computer science at the southwest normal University in 2006. Her research interests include digital image processing and virtual reality.