DETECTION OF IMBALANCED VERTICES IN 3D MESHES

Kaiyi Feng, Qi Li, Yongyi Gong, Jing Yang

Guangdong University of Foreign Studies
Cisco School of Informatics
Guangzhou, China

Silong Peng

Chinese Academy of Sciences
Institute of Automation
Beijing, China

ABSTRACT

A critical task in corner detection in 2D images is on the distinction a corner pixel and a pixel with a large gradient (i.e., an edge pixel). Imbalanced point detection was proposed to address this problem, where a corner pixel is characterized as a pixel with an imbalanced appearance, while an edge pixel has the opposite property. With extensive experiments, an imbalanced point detector has been shown to outperform many state-of-the-art detectors in terms of high repeatabilities and localization accuracy. A critical task in corner detection in 3D meshes is on the distinction of a corner vertex and a vertex with a large curvature. In this paper, we generalize the idea of imbalance oriented corner detection from 2D images to 3D meshes. We propose four algorithms on the detection of imbalanced vertices in 3D meshes. We present an experimental study of the four proposed algorithms and two curvature-based methods. Results show the promise of the proposed method on detection of 3D imbalanced vertices.

Index Terms— Mesh, normal vector, tangent plane, corner, imbalance.

1. INTRODUCTION

Corner detection aims to extract a compact (relatively small) set of 2D or 3D points that are repeatable with respect to various transforms of input data, in addition to having distinct appearances [6, 15, 5, 8]. There are some alternative terms for corner detection, such as interest point detection [15, 12] and keypoint detection [11]. A subtle difference between a corner point and an interest point (or a keypoint) is that a corner point does not have characteristic scales, while an interest point (or a keypoint) is essentially an interest region that is detected via the Lindeberg's scale-space theory [8]. In many vision applications under a constrained scaling setup (i.e., no significant scale variations), e.g., medical visual data registration [18, 1], border crossing inspection [3], corners without scale information can be straightly applied for matching visual data, such as 3D point clouds of faces of different persons. Given a vision application under an unconstrained scaling setup, we may apply a so-called by-pass strategy [8] to assign a scale to a corner by associating an interest region and the corner.

A critical task in corner detection in 2D images is on the distinction a corner pixel and a pixel with a large *gradient* (i.e., an edge pixel). *Harris detector* is a well-known corner detector, where the cornerness to an image pixel is decided by the product of the eigenvalues of the covariance matrix of gradients in addition to an additive term to penalize the edge response of the pixel [6, 15]. Imbalance oriented detection was proposed to distinguish a corner pixel and a pixel with a large gradient, where a corner pixel is characterized as a pixel with an imbalanced appearance, while an edge pixel has the opposite property. A distinctive property of an imbalance oriented point is that it has the so-called locality property, meaning that an imbalanced point may be a neighbor of some other imbalanced point [8]. The locality property of an imbalanced point is a direct outcome of an imbalanced point detector with one motivation of locating repeatable image pixels in sparsely-textured (or weakly-textured) images, such as human faces. More specifically, an imbalanced point detector applies a so-called imbalance oriented selection instead of the well-known *non-maximum suppression* to select candidates of repeatable image pixels since the latter one tends to over-suppress good candidates in sparsely-textured images. Moreover, non-maximum suppression can also degrade the localization accuracy of an interest point. With extensive experiments, imbalanced points have been shown to outperform many state-of-the-art detectors on 2D images in terms of the repeatability evaluation.

A critical task in corner detection in 3D meshes is on the distinction of a corner vertex and a vertex with a large curvature. Curvature is the fundamental property of a 3D point, which is parallel to the grayvalue of a 2D image pixel. Many interest point detectors in 3D meshes is based on curvatures [4, 16, 20, 19]. Based on curvatures, Sipiran and Bustos [16] generalized Harris detector from 2D images to 3D meshes; Zaharescu et al. [20] generalized the Difference of Gaussian

The work of Kaiyi Feng, Qi Li, and Yongyi Gong was supported by National Science Foundation Grant of China 61370160, Guangdong Province Natural Science Foundation Project 2015A030313578,Guangdong Scientific and Technological Plan Project 2015B010106005, Guangzhou Scientific and Technological Plan Project 2014J4100032. The corresponding author is Yongyi Gong.

(DOG) operator from 2D images [11] to 3D meshes. The most popular curvature used in interest point detection in 3D meshes is the Gaussian curvature [17, 20, 19]. Other types of curvatures were also studied in the context of 3D matching, e.g., normal curvature [14], mean curvatures [20], overall maximum curvatures [10], etc. Sun et al. [17] proposed a method named Heat Kernel Signature that is directly related to the Gaussian curvature on a surface for a small signature t. In general, the number of interest points detected by Heat Kernel Signature is small [17, 20].

Unlike the above-mentioned curvature based methods , Novatnack and Nishino [13] proposed a scale-dependent features, i.e., corner points, by analyzing the Gram matrix constructed on the normal map of a given 3D mesh. Bariya and Nishino [2] extended the work of Novatnack and Nishino [13] by building a model library of all objects that are to be recognized so that each object and scene can be represented by a set of scale-dependent corners and their scale-invariant local 3D shape descriptors.

In this paper, we generalize the idea of imbalanced pixels in 2D images to imbalanced vertices in 3D meshes. Similar to Novatanack and Nishino [13], we focus on the detection of corners and we use normal information instead of curvatures. Consistent with the previous work on imbalanced point detection [9, 7, 8], we do not apply non-maximum suppression to select good candidates. We propose four versions of implementation for the detection of imbalanced vertices in a 3D mesh. We present an experimental study of the four proposed implementation and two curvature-based methods. Results show the promise of the proposed method on detection of 3D imbalanced vertices.

The rest of the paper is organized as follows: In Section 2, we present a brief review on the detection of imbalanced pixels in 2D images. In Section 3, we propose four versions of generalization of imbalanced point detection in 3D meshes. We present experimental study in Section 4, and make conclusion in Section 5.

2. IMBALANCED PIXELS IN 2D IMAGES: A BRIEF REVIEW AND ABSTRACTION

In this section, we will give a brief review of the basic idea of imbalanced pixels in 2D images [9]. We will also give an abstraction of the algorithm on imbalance verification for the convenience of the generalization.

Imbalanced point detection in 2D images aims to minimize the occurrences of edge points [9]. Since edge points have similar local appearances (i.e., not distinctive to each other), they increase the chance of mismatching in the higher-level applications. Edge points can be characterized as points of balanced local appearances. Specifically, denote I a gray value image, p an image point, $\theta_i = (i-1) * 2\pi/N$, and $l_i = (\cos\theta_i, \sin\theta_i)$, for $i = 1, \ldots, N$. Denote $\frac{\partial I}{\partial l_i}(p)$ a directional derivative of p along l_i direction. We cluster $\{\frac{\partial I}{\partial l_i}(p)\}_{i=1}^{N}$ into two classes in terms of their magnitudes $|\frac{\partial I}{\partial l_i}(p)|$. If two clusters have the same size, the image point p is called *balanced*. Otherwise, p is *imbalanced*.

A sorting method [9] was proposed to group $\{|\frac{\partial I}{\partial l_i}(p)|\}_{i=1}^{N}$ into two classes. This idea can be generalized to deal with any scalar property of 3D data, such as cross angles of normal vectors of neighboring faces in a 3D mesh, etc. In general, given a set of non-negative real numbers $\{s_i\}_{i=1}^{N}$, we first sort these numbers in an increasing order such that

$$s_{i_1} \leq \ldots \leq s_{i_N},$$

where $i_j, j = 1, \ldots, N$, is a permutation of numbers $1, \ldots, N$. Denote

$$\text{maxDiff} = \max_{j=1,\ldots N-1}(s_{i_{j+1}} - s_{i_j}),$$

and denote the *index of maximum difference* as

$$j^* = \text{argmax}_{j=1,\ldots N-1}(s_{i_{j+1}} - s_{i_j}). \qquad (1)$$

If the index of maximum difference j^* is equal to $\frac{N}{2}$, p is considered as a balanced point. Otherwise, p is imbalanced. In [9], the authors further consider an image point p whose index of maximum difference is larger than $N/2$ as a redundant point because of its co-occurrence of a certain point whose index of maximum difference is less than $N/2$.

Fig. 1 illustrates the basic idea of imbalance oriented selection given $N = 8$. A long (short) arrow indicates a directional derivative on p of a large (small) magnitude. The number of long arrows is equal to the number of short arrows, which indicates the balance nature of an edge point. Fig. 1 (a) shows a case of a balanced point. Fig. 1 (b) shows a case of an imbalanced point, where the number of long arrows is not equal to the number of short arrows.

Next, we present two generic schemes on imbalance verification, which are basically two simple binary clustering methods. Alg. 1 presents a generic (an abstract) scheme on imbalance verification using the index of maximum of difference j^*. There are two important notes on this generic scheme.

- the index of maxDiff j^* is used to distinguish imbalanced and balanced points.
- maxDiff with a threshold on homogeneity T_h is used to identify a point with "homogeneous" appearance. Specifically, a point with a set of geometry values that has small variation is removed from the candidate set of imbalanced points.

It is also worth noting that Step 6 of Alg. 1, i.e., the condition check $j^* < N/2$, may be changed to $j^* > N/2$. Both conditions are subsets of $j^* \neq N/2$. A pixel p with $j^* < N/2$ (i.e., a pixel inside a corner neighborhood) is acompanied with some pixel with $j^* > N/2$ (i.e., a pixel outside the same corner neighborhood of p). In the context of corner detection

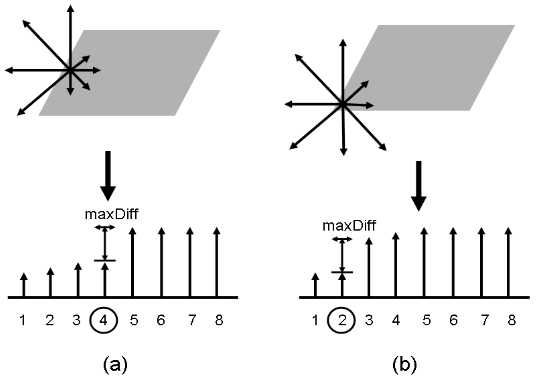


Fig. 1: Illustration of the imbalance oriented selection. Eight arrows are sorted in term of their magnitudes, as shown in the second row. (a) An edge point of balanced local appearance, where the index of maximum difference is 4 (half of 8 directions); (b) An imbalanced point, where the index of maximum difference is 2 ($\neq$ half of 8 directions).

in 2D images, $j^* < N/2$ can characterize a more compact set of corner candidates than $j^* > N/2$, as illustrated in Fig. 1 (b), which is in turn due to the geometric property of directional derivatives.

Algorithm 1 Imbalance verification based on j^*

```
input    $S = \{s_j\}_{j=1}^N$ // a set of geometry values
         $T_h$ : a threshold on homogeneity
output   imbalance: 1/0

1. sort $S$ in an increasing order
2. compute the maxDiff and $j^*$
3. if (maxDiff $< T_h$)
4.     imbalance = 0; // homogeneity case
5. else
6.     if $j^* < \frac{N}{2}$
7.         imbalance = 1;
8.     else
9.         imbalance = 0;
```

Alg. 2 presents an alternative method for imbalance verification. There are three important notes on this method:

- Unlike Alg. 1, Alg. 2 does not apply a sorting method for a binary clustering of a given set of geometry values, and thus it is more efficient.
- Alg. 2 involves a threshold T_c for a binary clustering.
- Alg. 2 does not have a threshold on homogeneity.

Similar to Alg. 1, Step 2 of Alg. 2 may be changed to num $> \frac{N}{2}$, depending on the set of geometry values.

Algorithm 2 Imbalance verification by thresholding

```
input    $S = \{s_j\}_{j=1}^N$ // a set of geometry values
         $T_c$ : a threshold for clustering
output   imbalance: 1/0

1. num = $|\{j : s_j > T_c\}|$
2. if num $< \frac{N}{2}$
3.     imbalance = 1;
4. else
5.     imbalance = 0;
```

3. IMBALANCED VERTICES IN 3D MESHES

In this section, we will generalize the characterization of imbalance of 2D pixels to 3D vertices by replacing the directional derivative operator by three types of geometric operators based on face normal vectors. We propose four methods for detection of imbalanced vertices in 3D meshes by integrating three types of geometric operators and two abovementioned generic imbalance verification schemes.

We first introduce a few notations for convenience of illustration, as shown in Table 1. Specifically, denote M a 3D mesh, v_i a vertex in a given 3D mesh, and $F_k(v_i)$ the set of all faces within the k-ring neighborhood of a vertex v_i. Usually, k is set to be 1 or 2. Denote n_f the normal vector of a face f, n_i the normal vector of the vertex v_i that is the average of normal vectors of all k-ring faces of $n_{f \in F_k(v_i)}$. Denote t_i the tangent plane of v_i, i.e., a plane perpendicular to the normal vector n_i. Denote $p_{f,i}$ the projection of the normal vector n_f on the tangent plane t_i. Given a 1-ring neighborhood of a vertex v_i, we order the faces within $F_1(v_i)$ in clockwise.

notation	meaning
M	a 3D mesh
v_i	a vertex in a mesh M
$F_k(v_i)$	all faces in the k-ring of v_i
f	a face in a mesh
n_f	the normal vector of face f
n_i	the normal vector of vertex v_i
t_i	tangent plane of v_i
$p_{f,i}$	projection of n_f on t_i
T_h	a threshold on homogeneity
T_c	a threshold for clustering

Table 1: Notations

Given a vertex v_i with a k-ring neighborhood $F_k(v_i)$, we investigate three types of 3D geometric operators for the characterization of the imbalance of the vertex v_i. These three types of geometric operators are all based on the normal vector (equivalently the tangent plane) of the vertex v_i, and the normal vectors of faces within a k-ring neighborhood of v_i.

Type-1 operator: cross angles between projections of face

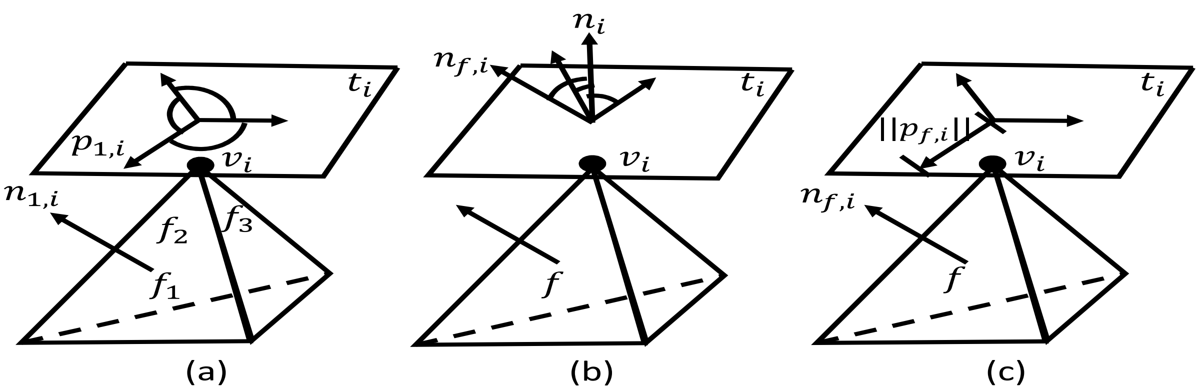


Fig. 2: Illustration of three types of geometric operators. (a) Type-1, (b) Type-2, and (c) Type-3.

normals, i.e.,

$$s_j = \text{cross angle between } p_{j,i} \text{ and } p_{j+1,i}, \quad (2)$$

where $p_{j,i}$ is the projection of the normal vector n_{f_j} on the tangent plane t_i. Type-1 operator is applicable to faces in 1-ring neighborhood $F_1(v_i)$, where the faces within $F_1(v_i)$ are ordered in clockwise.

Type-2 operator: cross angles between the normals of faces and the vertex, i.e.,

$$s_f = \text{cross angle between } n_f \text{ and } n_i, \quad (3)$$

where n_f is the normal vector of face f, and n_i is the normal vector of the vertex v_i. Type-2 operator is applicable to faces in any k-ring neighborhood, $k \geq 1$. For a corner vertex, we can expect that the majority of cross angles between normal vectors of faces within k-ring neighborhood and the normal vector of v_i is large.

Type-3 operator: lengths of projections of face normals, i.e.,

$$s_f = ||p_{f,i}||, \quad (4)$$

where $p_{f,i}$ is the projection of n_f on t_i. Similar to Type-2 operator, Type-3 operator is applicable to faces in any k-ring neighborhood, $k \geq 1$. Fig. 2 illustrates the basic ideas of the three types of operators.

Next, we propose four algorithms, Algs. 3-6 by combining three types of geometric operators and two binary clustering methods (as shown in Alg. 1 and Alg. 2). In Alg. 1, the alternative imbalance check in Step 6, i.e., $j^* > N/2$, is used; in Alg. 2, the alternative imbalance check in Step 2, i.e., num $> N/2$, is used.

Table 2 summarizes the main characteristics of proposed algorithms. Alg. 3-5 are based on the type-1, -2, and -3 geometric operators, respectively. They all apply the sorting-based clustering method (Alg. 1) for imbalance verification, and thus imbalanced vertices detected by these three algorithms are straightforward generalization of imbalanced pixels. As an exception, Alg. 6 apply the thresholding-based clustering method (Alg. 2) for imbalance verification. Alg. 3 is applicable to 1-ring neighborhood only since the computation of type 1 geometry values requires a spatial order on faces with a neighborhood of a given vertex, and only 1-ring neighborhood can support a meaningful spatial order on faces.

algorithm	geometric operator	clustering	k-ring
Alg. 3	Type-1	j^*	1
Alg. 4	Type-2	j^*	1 or 2
Alg. 5	Type-3	j^*	1 or 2
Alg. 6	Type-2	thresholding	1 or 2

Table 2: Summary of proposed algorithms

Algorithm 3 Imbalance verification: Type1+j^*

```
input   $M$: a 3D mesh
        $T_h$ : a threshold on homogeneity
output  imbalance($v_i$): 1/0

for each vertex $v_i \in M$
    compute $n_i$ // normal vector of $v_i$
    compute $t_i$ // tangent plane of $v_i$
    compute $\{s_j\}_{j=1}^{|F_1(v_i)|}$ // Type-1 operator (Eq. 2)
    imbalance($v_i$) = Algorithm 1($\{s_j\}_{j=1}^{|F_1(v_i)|}, T_h$)
```

4. EXPERIMENTS

In this section, we present an evaluation of the repeatability of imbalanced vertices detected by four proposed algorithms with a comparison of the repeatability of interest vertices of high curvatures detected by two methods. The two methods output vertices of 10% largest curvatures, where one method applies non-maximum suppression, and the other doesn't. For convenience, we denote *c1* the method using curvatures without non-maximum suppression, and *c2* the method using curvatures with non-maximum suppression. We have two tested meshes: i) Dragon with 25337 vertices, and ii) Armadillo with 17698 vertices.

Algorithm 4 Imbalance verification: Type2+j^*

```
input   $M$: a 3D mesh
        k : k-ring
        $T_h$ : a threshold on homogeneity
output  imbalance($v_i$): 1/0

for each vertex $v_i \in M$
    compute $n_i$ // normal vector of $v_i$
    compute $t_i$ // tangent plane of $v_i$
    compute $\{s_f | f \in F_k(v_i)\}$ // Type-2 operator (Eq. 3)
    imbalance($v_i$) = Algorithm $1(\{s_f | f \in F_k(v_i)\}, T_h)$
```

Algorithm 5 Imbalance verification: Type3+j^*

```
input   $M$: a 3D mesh
        k : k-ring
        $T_h$ : a threshold on homogeneity
output  imbalance($v_i$): 1/0

for each vertex $v_i \in M$
    compute $n_i$ // normal vector of $v_i$
    compute $t_i$ // tangent plane of $v_i$
    compute $\{s_f | f \in F_k(v_i)\}$ // Type-3 operator (Eq. 4)
    imbalance($v_i$) = Algorithm $1(\{s_f | f \in F_k(v_i)\}, T_h)$
```

Algorithm 6 Imbalance verification: Type2+thresholding

```
input   $M$: a 3D mesh
        k : k-ring
        $T_c$ : a threshold for clustering
output  imbalance($v_i$): 1/0

for each vertex $v_i \in M$
    compute $n_i$ // normal vector of $v_i$
    compute $t_i$ // tangent plane of $v_i$
    compute $\{s_f | f \in F_k(v_i)\}$ // Type-2 operator (Eq. 3)
    imbalance($v_i$) = Algorithm $2(\{s_f | f \in F_k(v_i)\}, T_c)$
```

The threshold T_c in Alg. 6 is set to 40. Recall that T_c is the threshold to group a set of geometry values that cross angles between the normal vectors of faces and the normal vectors of a vertex v_i into two clusters: one with large cross angles, and the other with small cross angles. $T_c = 40$ implies that Alg. 6 considers cross angles larger than or equal to 40 as large angles. We also tested other setting of T_c ranging from 40 to 50, and observed similar results. So T_c is not a sensitive parameter in Alg. 6.

4.1. Visual comparison

In this comparison, we set $k = 1$ in the four proposed algorithms. For simplicity, we do not apply the threshold T_h on homogeneity in Alg. 3, 4, and 5. In other words, we output all candidates of imbalanced vertices verified by these three algorithms. Note that T_h is expected to be consistent with different types of operators, and thus the involvement of thresholds on homogeneity can increase the uncertainty. Also recall that Alg. 6 does not apply the threshold on homogeneity.

Figs. 3 and 4 show results of six methods on Dragon and Armadillo 3D meshes, respectively. The numbers of vertices detected by different methods are listed in Table 3. We can observe that Algs 3-5 all detected a large number of imbalanced vertices that are mostly vertices with high curvatures. Alg. 6 detected a much smaller set of imbalanced vertices that are mostly corner vertices.

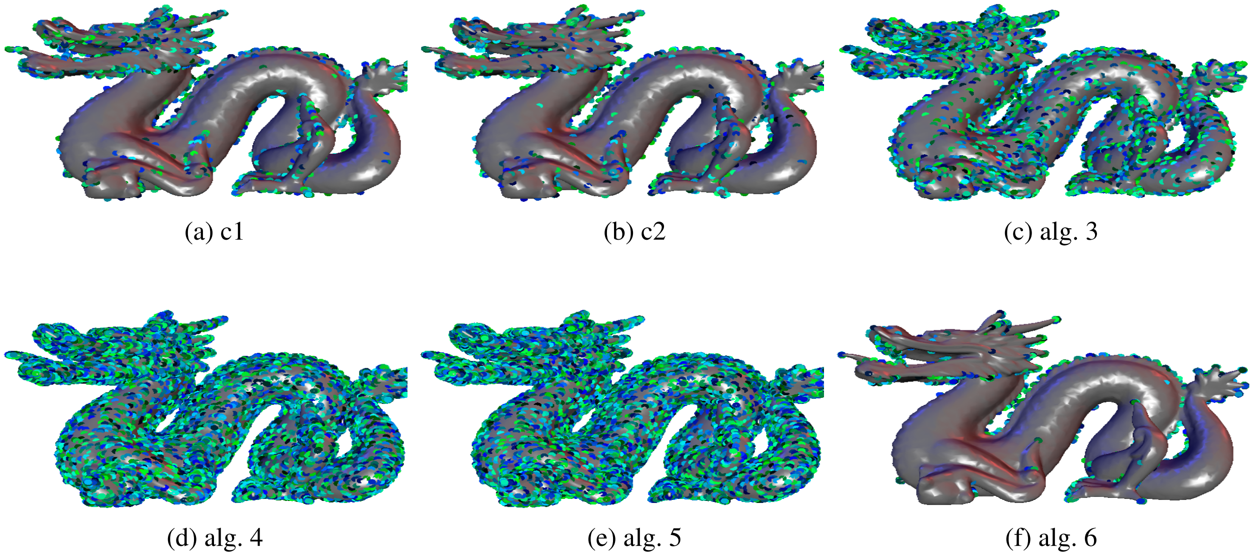
method	Dragon	Armadillo
c1	2407	1682
c2	1382	1099
alg. 3	6734	3815
alg. 4	14501	10207
alg. 5	15196	10705
alg. 6	1480	913

Table 3: Numbers of interested vertices detected in two meshes. Recall that c1 denotes the curvature method without non-maximum suppression, and c2 denotes the curvature method with non-maximum suppression.

4.2. Repeatability evaluation

In the following, we will evaluate repeatabilities of imbalanced vertices across three types of transformations: i) smoothing, ii) noise disturbance, and iii) scaling. We follow the definition of ϵ-repeatability rate proposed in [15] where ϵ is the tolerance in locating interest points under a transformation and set to be 1%.

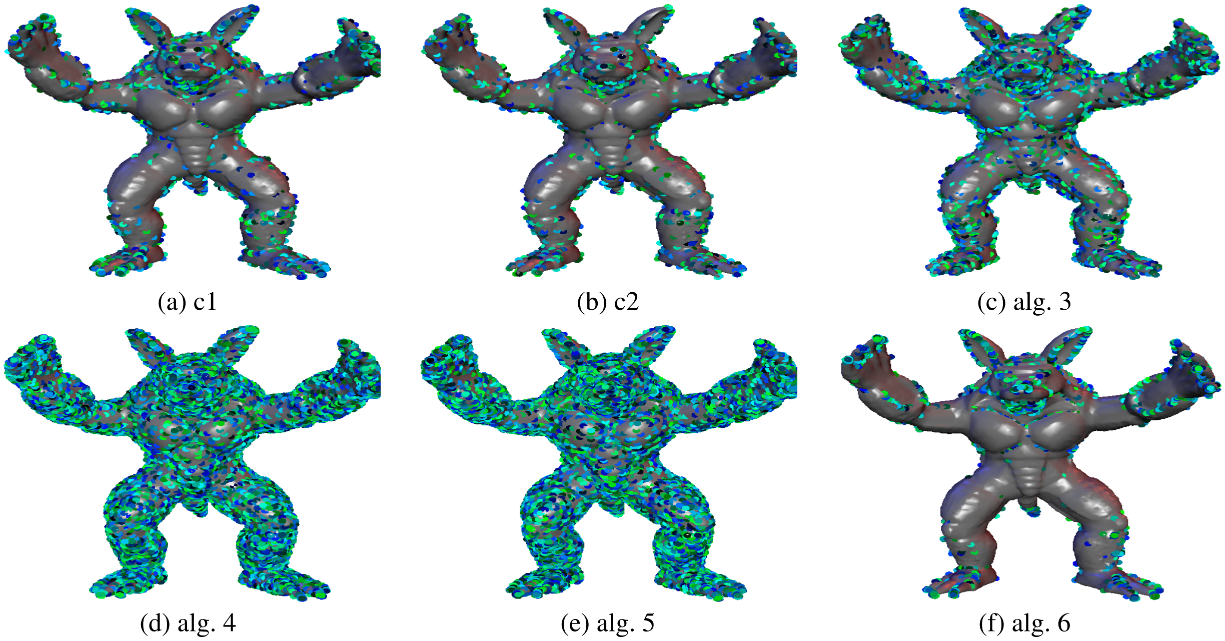
Results of repeatabilities across smoothing operations on Dragon and Armadillo are listed in Tabs. 4 and 5, respectively. We can observe that Alg. 6 outperforms all other methods consistently on two test meshes.



(a) c1 (b) c2 (c) alg. 3

(d) alg. 4 (e) alg. 5 (f) alg. 6

Fig. 3: Visual comparison on the Dragon 3D mesh.



(a) c1 (b) c2 (c) alg. 3

(d) alg. 4 (e) alg. 5 (f) alg. 6

Fig. 4: Visual comparison on the Armadillo 3D mesh.

Results of repeatabilities across noise on Dragon and Armadillo are listed in Tabs. 6 and 7, respectively. We can observe that c1 and Alg. 6 are two best methods: c1 performs relatively better in the cases of small noise (such as 1%), while Alg. 6 performs relatively better in the cases of large noise (such as 7%-10%). It is interesting to observe that the repeatability rate of imbalanced vertices detected by Alg. 6 increases rather than decreases as the noise level increases to 5% or higher. This is because Alg. 6 can detect a larger set of imbalanced vertices with a more noisy mesh, which in turn increases the number of matched vertices in two input meshes (i.e., the numerator of a repeatability rate) while not increasing the minimum of numbers of imbalanced vertices detected in two input meshes (i.e, the denominator of the repeatability rate).

Results of repeatabilities across scaling on Dragon and Armadillo are listed in Tabs. 8 and 9, respectively. We can observe that Alg. 3 and c1 are two best methods, and Alg. 3 performs better than c1 in the majority of test.

method	#iter=1	#iter=3	#iter=6
c1	0.87	0.75	0.67
c2	0.72	0.52	0.43
alg. 3	0.81	0.77	0.77
alg. 4	0.77	0.68	0.66
alg. 5	0.79	0.72	0.68
alg. 6 (k=1)	0.85	0.77	0.71
alg. 6 (k=2)	**0.88**	**0.84**	**0.81**

Table 4: Repeatability across smoothing operations on Dragon mesh. Alg. 6 achieves highest repeatabilities consistently.

method	#iter=1	#iter=3	#iter=6
c1	0.84	0.71	0.65
c2	0.73	0.57	0.48
alg. 3	0.77	0.73	0.73
alg. 4	0.73	0.66	0.64
alg. 5	0.76	0.69	0.66
alg. 6 (k=1)	0.86	0.74	0.68
alg. 6 (k=2)	**0.91**	**0.84**	**0.81**

Table 5: Repeatability across smoothing operations on Armadillo mesh. Alg. 6 achieves highest repeatabilities consistently.

5. CONCLUSION AND FUTURE WORK

In this paper, we generalized the basic idea of imbalanced pixels in 2D images to imbalanced vertices in 3D meshes for the detection of corner vertices. We studied three types of operators to construct a set of geometry values for each vertex, and

method	1%	3%	5%	7%	10%
c1	**0.95**	**0.86**	0.80	0.74	0.67
c2	0.90	0.77	0.66	0.57	0.48
alg. 3	0.86	0.75	0.70	0.69	0.68
alg. 4	0.85	0.73	0.71	0.69	0.69
alg. 5	0.87	0.76	0.73	0.74	0.75
alg. 6 (k=1)	0.88	0.81	0.82	0.83	0.85
alg. 6 (k=2)	0.91	0.85	**0.86**	**0.87**	**0.91**

Table 6: Repeatability across noise operations Dragon mesh. c1 and Alg. 6 are two best methods: c1 performs relatively better in the cases of small noise (1% - 3%), while Alg. 6 performs relatively better in the cases of large noise (5%-10%).

method	1%	3%	5%	7%	10%
c1	**0.95**	0.84	**0.77**	0.72	0.65
c2	0.91	0.79	0.66	0.62	0.52
alg. 3	0.87	0.77	0.71	0.67	0.64
alg. 4	0.86	0.75	0.70	0.69	0.68
alg. 5	0.87	0.77	0.73	0.71	0.72
alg. 6 (k=1)	0.89	0.78	0.73	0.71	0.71
alg. 6 (k=2)	0.91	**0.85**	**0.77**	**0.78**	**0.80**

Table 7: Repeatability across noise operations on Armadillo mesh. c1 and Alg. 6 are two best methods: c1 performs better in the cases of small noise (such as 1%), while Alg. 6 performs better in the cases of large noise (such as 7%-10%).

method	(1.1,1,1)	(1.3,1,1)	(1.5,1,1)
c1	0.95	0.87	0.81
c2	0.93	0.82	0.73
alg. 3	**0.97**	**0.93**	**0.90**
alg. 4	0.95	0.89	0.85
alg. 5	0.96	0.90	0.86
alg. 6 (k=1)	0.95	0.89	0.85
alg. 6 (k=2)	0.95	0.89	0.84

Table 8: Repeatability across scaling operations on Dragon mesh. Alg. 3 and c1 are two best methods, and Alg. 3 performs better than c1.

method	(1.1,1,1)	(1.3,1,1)	(1.5,1,1)
c1	**0.95**	0.85	0.76
c2	0.92	0.80	0.69
alg. 3	0.91	**0.88**	**0.86**
alg. 4	0.90	0.86	0.83
alg. 5	0.90	0.87	0.84
alg. 6 (k=1)	0.92	0.82	0.76
alg. 6 (k=2)	0.91	0.82	0.75

Table 9: Repeatability across scaling operations on Armadillo mesh. Alg. 3 and c1 are two best methods, and Alg. 3 performs better than c1.

we proposed four algorithms for the detection of imbalanced vertices, where three of them applied a sorting-based scheme for the binary clustering task in imbalance verification that was proposed to detect imbalanced pixels in 2D images, and one of them applied a thresholding scheme for the binary clustering task. The experiment results showed that the algorithm built up with the type-2 operator and the thresholding-based clustering method, i.e., Alg. 6, detected a compact set of imbalanced vertices that are consistent with corner vertices, and achieved the overall best performance on the evaluation of repeatabilities across smoothing and noise. In the future, we will study representation/descriptors of imbalanced vertices for mesh matching or registration.

6. REFERENCES

[1] L. I. Abe, Y. Iwao, T. Gotoh, S. Kagei, R. Y. Takimoto, M. S. G. Tsuzuki, and T. Iwasawa. High-speed point cloud matching algorithm for medical volume images using 3d voronoi diagram. In *7th International Conference on Biomedical Engineering and Informatics, BMEI 2014, Dalian, China, October 14-16, 2014*, pages 205–210, 2014.

[2] P. Bariya and K. Nishino. Scale-hierarchical 3d object recognition in cluttered scenes. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 1657–1664, 2010.

[3] C. Busch and A. Nouak. 3d face recognition for unattended border control. In *Proceedings of the 2008 International Conference on Security & Management, SAM 2008, Las Vegas, Nevada, USA, July 14-17, 2008*, pages 350–356, 2008.

[4] C. Dorai and A. Jain. COSMOS - A representation scheme for 3d free-form objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(10):1115–1130, 1997.

[5] Y. Guo, M. Bennamoun, F. A. Sohel, M. Lu, and J. Wan. 3d object recognition in cluttered scenes with local surface features: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2270–2287, 2014.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conference, Manchester*, pages 147–151, 1988.

[7] Q. Li. Interest points of general imbalance. *IEEE Transactions on Image Processing*, 18(11):2536–2546, 2009.

[8] Q. Li and Y. Gong. Scale invariant representation of imbalanced points. *Neurocomputing*, 173:1422–1435, 2016.

[9] Q. Li, J. Ye, and C. Kambhamettu. Interest point detection using imbalance oriented selection. *Pattern Recognition*, 41(2):672–688, 2008.

[10] F. Liu, D. Zhang, and L. Shen. Study on novel curvature features for 3d fingerprint recognition. *Neurocomputing*, 168:599–608, 2015.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *IEEE International Conference on Computer Vision*, volume I, pages 525–531, Vancouver, Canada, 2001.

[13] J. Novatnack and K. Nishino. Scale-dependent 3d geometric features. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, pages 1–8, 2007.

[14] J. Novatnack and K. Nishino. Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images. In *Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part III*, pages 440–453, 2008.

[15] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[16] I. Sipiran and B. Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963–976, 2011.

[17] J. Sun, M. Ovsjanikov, and L. J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Comput. Graph. Forum*, 28(5):1383–1392, 2009.

[18] G. K. L. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, A. D. Marshall, R. R. Martin, X. Sun, and P. L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Trans. Vis. Comput. Graph.*, 19(7):1199–1217, 2013.

[19] F. Tombari, S. Salti, and L. di Stefano. Performance evaluation of 3d keypoint detectors. *International Journal of Computer Vision*, 102(1-3):198–220, 2013.

[20] A. Zaharescu, E. Boyer, and R. Horaud. Keypoints and local descriptors of scalar functions on 2d manifolds. *International Journal of Computer Vision*, 100(1):78–98, 2012.