



Department of Informatique,  
Faculty of Sciences and Techniques,  
Université Abdelmalek Essaâdi. Tanger, Morocco



---

## Devoir individuel II: Réduction de maillage

---

MST: Intelligence Artificielle et Sciences de Données

Préparé par:  
FAHD CHIBANI

Encadré par  
Pr. M'hamed AIT KBIR

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Objectif</b>	<b>2</b>
<b>3</b>	<b>Théorie et Concepts de Base</b>	<b>2</b>
3.1	Description du Format OBJ . . . . .	2
3.2	Présentation des Algorithmes . . . . .	3
3.2.1	Vertex Clustering . . . . .	3
3.2.2	Edge Collapse . . . . .	4
3.3	Critères de Comparaison . . . . .	6
<b>4</b>	<b>Méthodologie</b>	<b>6</b>
4.1	Décodage des Fichiers OBJ . . . . .	6
4.2	Implémentation de l'Algorithme de Vertex Clustering . . . . .	6
4.3	Implémentation de l'Algorithme d'Edge Collapse . . . . .	7
<b>5</b>	<b>Expérimentations et Résultats</b>	<b>9</b>
5.1	Présentation des modèles de test . . . . .	9
5.2	Interprétation des résultats obtenus . . . . .	10
<b>6</b>	<b>Conclusion</b>	<b>12</b>
<b>7</b>	<b>References</b>	<b>13</b>

# 1 Introduction

La réduction de maillage 3D est une étape essentielle dans de nombreux domaines de l'infographie et de la vision par ordinateur. Elle consiste à simplifier les modèles 3D en réduisant le nombre de sommets et de faces tout en maintenant un niveau acceptable de qualité visuelle. Cette simplification permet notamment d'optimiser les performances, de réduire l'espace de stockage nécessaire et d'accélérer les traitements tels que le rendu graphique ou l'analyse.

Dans le cadre de ce projet, nous explorons deux algorithmes de réduction de maillage : le *Vertex Clustering* et le *Edge Collapse*. Ces méthodes visent à produire des modèles 3D simplifiés tout en minimisant la perte des détails visuels importants.

## 2 Objectif

L'objectif principal de ce projet est de développer une solution permettant de réduire le nombre de sommets et de faces d'un modèle 3D tout en préservant la qualité visuelle et les détails. Pour ce faire, les étapes suivantes seront réalisées :

- Décoder un fichier au format `.obj` et mettre en place des structures de données adaptées pour représenter les sommets, les arêtes et les faces du modèle.
- Implémenter les algorithmes de réduction de maillage *Vertex Clustering* et *Edge Collapse*.
- Sauvegarder et comparer les résultats obtenus à l'aide des modèles fournis dans le sous-dossier `DVI2`, en analysant la réduction du nombre de sommets et de faces, ainsi que la qualité visuelle et les temps d'exécution.

Ce projet offre une opportunité de comprendre les principes fondamentaux de la simplification des maillages 3D et de les mettre en pratique dans un cadre algorithmique.

## 3 Théorie et Concepts de Base

La réduction de maillage 3D repose sur des principes fondamentaux de la géométrie et de l'infographie. Avant d'implémenter les algorithmes de simplification, il est essentiel de comprendre la structure des fichiers `.obj`, qui représentent les modèles 3D, ainsi que les concepts associés aux techniques de réduction. Cette section présente les éléments clés nécessaires à la compréhension et à l'implémentation des algorithmes.

Nous commencerons par une description du format Wavefront OBJ, un standard largement utilisé pour stocker des modèles 3D. Ensuite, nous détaillerons les principes des deux algorithmes de réduction étudiés dans ce projet : *Vertex Clustering* et *Edge Collapse*. Enfin, nous introduirons les critères qui seront utilisés pour comparer les performances et les résultats des deux méthodes.

### 3.1 Description du Format OBJ

Le format Wavefront OBJ est un format de fichier simple et flexible, utilisé pour représenter des modèles 3D. Il se compose de différentes primitives qui définissent la géométrie du modèle :

- **Sommets (v)** : Ils représentent les points dans l'espace 3D, définis par leurs coordonnées  $(x, y, z)$ .
- **Faces (f)** : Elles définissent les surfaces du modèle en reliant les sommets. Une face est spécifiée par une liste d'indices faisant référence aux sommets.
- **Normales (vn)** : Ce sont des vecteurs qui indiquent la direction perpendiculaire à une surface, utiles pour l'éclairage et le rendu.
- **Coordonnées de texture (vt)** : Ces coordonnées servent à mapper une texture sur la surface du modèle.

Un exemple de fichier .obj est présenté ci-dessous :

```
1 v 0.13351 0.4321 -0.42496
2 v 0.19961 0.4512 -0.35165
3 v 0.21282 0.40178 -0.42255
4 ...
5 vt 0.6165 2.1421
6 vt 0.6165 2.1204
7 vt 0.6406 2.1182
8 ....
9 vn -0.330983 0.819014 0.468685
10 vn -0.378962 0.917356 0.121842
11 vn -0.380645 0.916596 -0.122319
12 ...
13 f 1/1/1 2/2/2 3/3/3
14 f 4/4/4 1/1/1 3/3/3
15 f 5/5/5 6/6/6 2/2/2
16 ...
```

## 3.2 Présentation des Algorithmes

Dans cette section, nous présentons deux approches principales pour la réduction de maillage : *Vertex Clustering* et *Edge Collapse*. Ces algorithmes permettent de réduire le nombre de sommets et de faces tout en essayant de maintenir la qualité visuelle du modèle.

### 3.2.1 Vertex Clustering

Le *Vertex Clustering* est une méthode largement reconnue pour la simplification des modèles 3D, essentielle en infographie et en modélisation. Cette technique regroupe les sommets à l'intérieur d'une grille spatiale définie, permettant une réduction efficace de la complexité du maillage tout en tentant de préserver la forme globale et les caractéristiques importantes du modèle original.

#### Principes de Base du Clustering des Sommets :

1. **Partitionnement de la Grille** : Le maillage original est encapsulé dans une boîte englobante (*bounding box*), divisée en une grille uniforme. Chaque cellule de cette grille peut contenir un ou plusieurs sommets du maillage.
2. **Agrégation des Sommets** : Les sommets situés dans une même cellule de la grille sont regroupés et remplacés par un sommet représentatif. Ce sommet représentatif est généralement choisi en fonction de critères spécifiques, tels que la proximité ou la courbure.
3. **Mise à Jour du Maillage** : Après le regroupement, les faces du maillage sont mises à jour pour refléter les nouveaux sommets représentatifs, réduisant ainsi le nombre total de sommets et de faces dans le modèle.

#### Avantages et Limitations :

##### Avantages

- **Efficacité** : Le *Vertex Clustering* est réputé pour son efficacité et sa rapidité dans la simplification des maillages, ce qui le rend adapté aux applications en temps réel.
- **Simplicité** : Cet algorithme est relativement simple à implémenter comparé à des méthodes plus complexes, comme la contraction des arêtes ou les techniques préservant la topologie.

## Limitations

- **Qualité du Résultat** : Un inconvénient majeur est que le *Vertex Clustering* peut entraîner une perte de détails, notamment dans les zones à forte courbure ou aux caractéristiques complexes. Le processus de voxelisation peut ne pas préserver adéquatement ces détails.
- **Dépendance à la Résolution de la Grille** : La qualité de la simplification dépend fortement de la résolution de la grille. Une grille grossière peut trop simplifier le maillage, tandis qu'une grille fine peut entraîner un traitement moins efficace sans apporter de gains significatifs en termes de préservation des détails.

## Avancées Récentes :

Des études récentes ont cherché à améliorer les techniques traditionnelles de *Vertex Clustering* en intégrant des critères supplémentaires pour la sélection des sommets représentatifs. Par exemple, des méthodes basées sur principal curvature ont été proposées pour améliorer la fidélité des modèles simplifiés en veillant à ce que les caractéristiques géométriques importantes soient préservées pendant la simplification. En outre, des implémentations sur GPU ont été explorées pour tirer parti des capacités de traitement parallèle, augmentant ainsi l'efficacité de la simplification tout en maintenant une qualité élevée.

### 3.2.2 Edge Collapse

Edge Collapse est une technique largement utilisée pour simplifier les modèles 3D en réduisant le nombre de sommets et d'arêtes tout en préservant la forme et l'apparence globale du modèle. Cette méthode est particulièrement utile dans des domaines tels que l'infographie, l'impression 3D et le rendu en temps réel, où les performances et l'efficacité mémoire sont essentielles.

#### i. Mécanisme d'Edge Collapse :

Le processus de contraction d'arêtes consiste à fusionner deux sommets adjacents reliés par une arête, éliminant ainsi cette arête du maillage. Les étapes typiques incluent :

##### - Sélection de l'Arête

Une arête est choisie pour être contractée sur la base d'une fonction de coût qui évalue l'impact sur la topologie du modèle. Les critères couramment utilisés incluent:

- **Distorsion Géométrique:** Minimiser les changements de forme.
- **Préservation des Caractéristiques:** Conserver des éléments importants tels que les arêtes ou les coins.
- **Fonctions de Coût:** Diverses fonctions de coût peuvent être définies, basées sur la courbure des sommets ou les propriétés locales de la surface.

##### - Suppression des Triangles

Les triangles partageant l'arête sélectionnée sont supprimés du maillage. Cette étape réduit la complexité du modèle tout en veillant à mettre à jour correctement les triangles adjacents.

##### - Mise à Jour des Sommets

Les triangles restants, qui utilisaient l'un des sommets contractés, sont mis à jour pour utiliser l'autre sommet à la place. Cela garantit la continuité dans la structure du maillage.

##### - Suppression des Sommets

Enfin, le sommet qui faisait partie de l'arête contractée est supprimé du maillage, complétant ainsi l'étape de simplification.

## ii. Métriques d'Erreur dans la Contraction d'Arêtes:

Pour garantir que la simplification préserve les caractéristiques importantes du modèle, différentes métriques d'erreur sont utilisées :

### - Mesure Précise d'Erreur Quadratique (AMQE)

Cette métrique évalue la déviation géométrique en calculant les distances au carré pondérées entre les surfaces simplifiées et originales du maillage. Elle aide à préserver les détails lors de la simplification :

$$E(v) = \sum_{f \in F} w_f d_f^2(v)$$

où  $E(v)$  est l'erreur associée au sommet  $v$ ,  $F$  est l'ensemble des faces,  $w_f$  est un poids pour la face  $f$ , et  $d_f(v)$  est la distance du sommet  $v$  à la face  $f$ .

### - Mesure Symétrique d'Erreur Quadratique (SMQE)

Cette méthode mesure l'erreur dans les deux directions (du maillage simplifié à l'original et vice versa), aidant à préserver les frontières et les arêtes vives, mais elle est plus coûteuse en calcul que l'AMQE:

$$E_{sym}(v) = E(v) + E(v_{original})$$

## iii. Défis et Bonnes Pratiques :

Bien que la contraction d'arêtes soit efficace, elle présente certains défis :

### Artefacts Non-Manifold

Il est crucial de s'assurer que la contraction d'une arête ne crée pas de géométries non-manifold, qui pourraient entraîner des problèmes de rendu ou des erreurs lors de l'impression 3D.

### Préservation des Caractéristiques

Il est essentiel de conserver les caractéristiques géométriques importantes pendant la simplification, en particulier pour les modèles où des détails spécifiques sont essentiels à la fidélité visuelle ou à l'intégrité fonctionnelle.

### Bonnes Pratiques :

- **Choisir des Méthodes Appropriées:** En fonction du type de modèle (par exemple, organique vs géométrique).
- **Utiliser des Techniques Adaptatives:** Ajuster les détails en fonction de la distance ou de l'angle de vue.
- **Mettre en Œuvre des Vérifications de Connectivité:** Pour éviter les artefacts pendant la contraction d'arêtes.
- **Approches Multi-résolutions:** Permettre différents niveaux de détail en fonction des besoins de l'application.
- **Contrôle Utilisateur:** Offrir des options permettant aux utilisateurs de spécifier les zones où les détails doivent être conservés.

### 3.3 Critères de Comparaison

Pour évaluer la performance des deux algorithmes, plusieurs critères seront pris en compte :

- **Réduction du nombre de sommets et de faces** : Mesure de la simplification obtenue.
- **Qualité visuelle** : Évaluation subjective ou objective des artefacts introduits par la simplification.
- **Temps d'exécution** : Comparaison de l'efficacité des algorithmes.
- **Impact géométrique** : Analyse quantitative, comme la distance de Hausdorff, pour mesurer la déviation entre le modèle initial et le modèle simplifié.

## 4 Méthodologie

Cette section décrit la méthodologie adoptée pour la simplification des modèles 3D. Elle se divise en deux parties principales: le décodage des fichiers `.obj` et l'implémentation des algorithmes de réduction de maillage *Vertex Clustering* et *Edge Collapse*.

### 4.1 Décodage des Fichiers OBJ

Le décodage des fichiers OBJ est une étape essentielle qui consiste à lire et interpréter les informations contenues dans ces fichiers pour extraire les primitives géométriques du modèle 3D. Cela inclut les sommets, les normales, les coordonnées de texture et les faces. Cette tâche est réalisée par la fonction `read_obj`, qui est responsable du parsing des fichiers OBJ et du stockage des données dans des structures adaptées.

**Processus de décodage:**

1. **Sommets (v)** : Les coordonnées  $(x, y, z)$  de chaque sommet sont extraites et stockées sous forme de tuples.
2. **Coordonnées de texture (vt)** : Les coordonnées  $(u, v)$  sont enregistrées pour mapper des textures sur les surfaces du modèle.
3. **Normales (vn)** : Les normales  $(nx, ny, nz)$ , utilisées pour l'éclairage, sont également extraites.
4. **Faces (f)** : Chaque face est représentée par une liste d'indices faisant référence aux sommets, textures et normales correspondants.

**Structures de données utilisées:** Les données extraites sont organisées dans un dictionnaire Python, où chaque clé représente un type de primitive:

- `'vertices'` : Liste des sommets.
- `'textures'` : Liste des coordonnées de texture.
- `'normals'` : Liste des normales.
- `'faces'` : Liste des faces, où chaque face est une liste de tuples représentant les indices des sommets, textures et normales.

Le code correspondant à cette partie se trouve dans le fichier `read_obj.py`.

### 4.2 Implémentation de l'Algorithme de Vertex Clustering

L'algorithme de *Vertex Clustering* simplifie le maillage en regroupant les sommets dans une grille spatiale régulière, où chaque cellule est remplacée par un sommet représentatif. Ce processus est implémenté dans le fichier `vertex_clustering.py`.

### Étapes principales:

1. **Création de la grille:** Les sommets du maillage sont projetés dans une grille tridimensionnelle, où chaque cellule correspond à un espace de coordonnées défini par la taille de la grille (`grid_size`). Cette opération est réalisée par la fonction `create_grid`.
2. **Calcul des sommets représentatifs:** Pour chaque cellule de la grille, un sommet représentatif est calculé en faisant la moyenne des coordonnées de tous les sommets qu'elle contient. Cela est implémenté dans la fonction `compute_representative_vertex`.
3. **Mise à jour des faces:** Les faces du maillage sont mises à jour pour utiliser les indices des nouveaux sommets représentatifs. La fonction vérifie également que les faces restent valides (au moins trois sommets distincts).
4. **Génération du nouveau maillage:** Un nouveau dictionnaire contenant les sommets, normales, textures et faces simplifiés est créé.

### Paramètres clés:

- `grid_size`: Contrôle le niveau de simplification. Une valeur plus grande conduit à une simplification plus importante.
- `old_to_new_index`: Une table de correspondance entre les anciens indices de sommets et les nouveaux indices.

**Sortie de l'algorithme:** L'algorithme retourne un dictionnaire contenant:

- `'vertices'` : La liste des nouveaux sommets simplifiés.
- `'faces'` : La liste des nouvelles faces mises à jour.
- `'textures'` et `'normals'` : Inchangées par rapport au fichier original.

Le résultat final peut être sauvegardé dans un fichier OBJ à l'aide de la fonction `write_obj`.

**Exemple d'exécution:** Voici un exemple d'utilisation de l'algorithme:

```
1 input_path = 'Obj/obj1.obj'
2 output_path = 'simplified.obj'
3 grid_size = 0.1 # Taille de la grille
4 obj_data = read_obj(input_path)
5 simplified_data = vertex_clustering(obj_data, grid_size)
6 write_obj(simplified_data, output_path)
```

**Statistiques générées:** L'algorithme produit des statistiques sur le maillage avant et après simplification:

- Nombre de sommets et de faces avant et après simplification.
- Détails sur les cellules de la grille utilisées.

## 4.3 Implémentation de l'Algorithme d'Edge Collapse

L'algorithme Edge Collapse est une technique de simplification de maillage qui réduit itérativement la complexité d'un modèle 3D tout en préservant sa forme générale. Notre implémentation permet de réduire le nombre de faces d'un maillage selon un ratio spécifié tout en minimisant la distorsion géométrique.



## Structure des données

Le maillage est représenté par une structure de données contenant :

- **vertices** : Liste des sommets en coordonnées 3D
- **faces** : Liste des faces triangulaires
- **normals** : Vecteurs normaux aux faces
- **textures** : Coordonnées de texture

## Algorithme principal

L'algorithme se décompose en plusieurs étapes clés :

### Initialisation

- 1: Calcul du nombre cible de faces :  $target_{faces} = faces_{initiales} \times ratio$
- 2: Création d'une file de priorité pour stocker les arêtes à effondrer
- 3: Identification de toutes les arêtes du maillage et leurs faces adjacentes

### Sélection et effondrement des arêtes

Pour chaque arête candidate à l'effondrement :

- Calcul du coût : Le coût d'effondrement est calculé comme la distance euclidienne entre les sommets :

$$[cost = \sqrt{\sum_{i=0}^2 (v1_i - v2_i)^2}]$$

**Processus d'effondrement :**

- Calcul du point médian entre les deux sommets
- Déplacement du premier sommet vers ce point médian
- Suppression des faces contenant le second sommet
- Mise à jour des références dans les faces restantes –

### Optimisations importantes

- **File de priorité** : Utilisation de `heapq` pour sélectionner efficacement les arêtes à effondrer
- **Gestion des références** : Maintien de la cohérence topologique lors des effondrements
- **Mise à jour incrémentale** : Actualisation efficace des coûts après chaque effondrement

### Fonction de coût

La fonction de coût implémentée est relativement simple mais efficace :

$$cost(edge) = |v_1 - v_2|_2 \tag{1}$$

où  $v_1$  et  $v_2$  sont les positions des sommets de l'arête.

### Conservation de la topologie

L'algorithme maintient la cohérence topologique du maillage en :

- Préservant la connectivité des faces
- Gérant correctement les références aux sommets
- Évitant la création de faces dégénérées

## Limitations et perspectives d'amélioration

- La fonction de coût pourrait être enrichie pour prendre en compte :
  - La préservation des caractéristiques géométriques
  - La courbure locale
  - Les attributs de surface (normales, textures)
- L'implémentation actuelle ne garantit pas la préservation optimale du volume
- Une parallélisation pourrait être envisagée pour les grands maillages

## 5 Expérimentations et Résultats

### 5.1 Présentation des modèles de test

Cette partie présente une analyse des modèles 3D fournis dans le dossier DVI2. Les fichiers analysés sont au format OBJ. Les informations extraites incluent le nombre de sommets (vertices), le nombre de normales (normals), le nombre de coordonnées de texture et le nombre de faces.

Fichier obj1.obj :

- **Sommets (vertices)** : 1369
- **Normales (normals)** : 1393
- **Coordonnées de texture** : 0
- **Faces** : 2734

Fichier obj2.obj :

- **Sommets (vertices)** : 10 505
- **Normales (normals)** : 10 535
- **Coordonnées de texture** : 13 855
- **Faces** : 20 822

Fichier obj3.obj :

- **Sommets (vertices)** : 2165
- **Normales (normals)** : 2167
- **Coordonnées de texture** : 0
- **Faces** : 4438

Fichier obj4.obj :

- **Sommets (vertices)** : 17 403
- **Normales (normals)** : 17 519
- **Coordonnées de texture** : 21 631
- **Faces** : 34 643

Fichier obj5.obj :

- Sommets (vertices) : 1 932
- Normales (normals) : 2 000
- Coordonnées de texture : 1 058
- Faces : 3 824

Fichier obj6.obj :

- Sommets (vertices) : 327 323
- Normales (normals) : 0
- Coordonnées de texture : 0
- Faces : 654 666

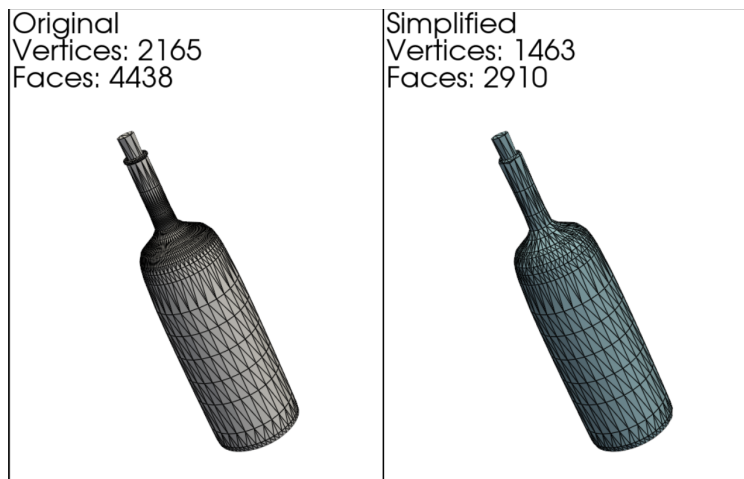
## 5.2 Interprétation des résultats obtenus




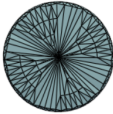
Interprétation des résultats du Vertex Clustering :

L'application du Vertex Clustering a permis de réduire efficacement le nombre de sommets et de faces du modèle 3D. Comme le montrent les figures, les résultats obtenus sur l'objet 6 du dossier DVI2 sont très satisfaisants. Nous avons atteint une réduction de 32,4 pourcent de la complexité du maillage, tout en préservant la forme globale et les caractéristiques essentielles du modèle d'origine.

```
PS C:\Users\lenovo\OneDrive\Desktop\AISD\S3\A
Simplified model saved to: simplified.obj



Reduction Statistics:
Vertices: 2165 → 1463 (32.4% reduction)
Faces: 4438 → 2910 (34.4% reduction)
```



Original Vertices: 2165 Faces: 4438	Simplified Vertices: 1463 Faces: 2910
	
	

### Interprétation des résultats d'Edge Collapse :

Après l'application de la méthode Edge Collapse pour réduire le nombre de faces des modèles 3D, on constate que les résultats restent satisfaisants, comme le montrent les figures suivantes. L'Edge Collapse a été appliqué à l'objet 6 du dossier DVI2, avec une réduction de 30 pourcent. Cette méthode a permis une réduction efficace de la complexité du maillage, tout en maintenant la forme globale et les caractéristiques essentielles du modèle original, sans erroné.

Original Model	Simplified Model
	



Même après avoir modifié le pourcentage de réduction des sommets dans l’algorithm d’Edge Collapse, les résultats obtenus restent inférieurs à ceux produits par la méthode de Vertex Clustering. En revanche, l’approche il a démontré une meilleure performance dans les scénarios où la conservation des caractéristiques visuelles et géométriques du modèle est primordiale. En supprimant de manière ciblée les arêtes tout en ajustant les positions des sommets voisins, cette méthode permet une réduction progressive et localisée de la complexité du maillage. Cela garantit une simplification plus équilibrée, préservant au mieux la forme globale et les particularités importantes du modèle original.

## 6 Conclusion

Dans ce rapport, nous avons étudié en détail deux approches fondamentales pour la simplification des maillages 3D : le Vertex Clustering et le Edge Collapse. Ces techniques ont été analysées pour évaluer leur capacité à réduire efficacement la complexité géométrique d’un modèle tout en maintenant un compromis entre la simplification et la préservation des détails visuels essentiels.

Le Vertex Clustering s’est distingué par son aptitude à réduire de manière significative le nombre de sommets et de faces. Cette méthode regroupe les sommets d’un maillage en clusters en fonction de leur proximité spatiale, permettant ainsi de diminuer rapidement la densité du maillage. En revanche, l’approche Edge Collapse a démontré une meilleure performance dans les scénarios où la conservation des caractéristiques visuelles et géométriques du modèle est primordiale.

Ces deux méthodes offrent des avantages complémentaires, le choix de l’une ou de l’autre dépendant principalement des exigences spécifiques de l’application, qu’il s’agisse de réduire drastiquement la complexité ou de maintenir la fidélité visuelle du modèle 3D.

## 7 References

1. [Mesh Simplification Using Vertex Clustering Based on Principal Curvature](#)
2. [Model Simplification Using Vertex-Clustering](#)
3. [A Novel Mesh Simplification Method Based on Vertex Removal Using Surface Angle](#)
4. [Mesh Simplification: Edge Collapse Conditions](#)
5. [Semantic Edge Collapse:A Mesh Edge Collapse Algorithm preserving per Face Semantic Information](#)