

# Сравнение многогранников: оценка качества алгоритма построения объёмной модели трехмерного тела

Попов И. В.

2 мая 2025 г.

## Аннотация

Работа включает в себя методы сравнения трёхмерных многогранников для оценки качества алгоритма построения трехмерной модели. Предложена система метрик между гранями трехмерных многогранников для определения расстояния между ними и дальнейшего его использования в различных алгоритмах нахождения оптимального распределения : венгерский алгоритм и жадный алгоритм.

## 1 Введение

### 1.1 Постановка проблемы

Алгоритму необходимо выдать число, которой характеризовало бы схожесть двух многогранников в некотором смысле. Сложности, которые возникают при попытке сравнения двух многогранников:

- Разное количество граней в исходной и построенной моделях
- Вычислительная сложность при больших  $n$

## 2 Математическая модель

### 2.1 Формальное описание многогранника

Многогранник  $P$  может быть задан в виде троки:

$$P = (V, E, F), \text{ где} \tag{1}$$

- $V = \{\mathbf{v}_i\} \subset \mathbb{R}^3$  — множество вершин
- $E = \{e_{ij}\}$  — множество рёбер
- $F = \{f_k\}$  — множество граней

Для каждой грани  $f_k$  многогранника определим её основные характеристики:

- Центр масс:  $\mathbf{c}_k = \frac{1}{S_k} \sum_{T \in \mathcal{T}_k} S_T \cdot \mathbf{c}_T$  где:

—  $\mathcal{T}_k$  — множество треугольников триангуляции грани  $f_k$  с помощью диагоналей

- $T$  – отдельный треугольник в триангуляции (элемент  $\mathcal{T}_k$ )
- $S_T = \frac{1}{2} \|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)\|$  – площадь треугольника  $T$  с вершинами  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$
- $\mathbf{c}_T = \frac{1}{3}(\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3)$  – центр масс треугольника  $T$
- $S_k = \sum_{T \in \mathcal{T}_k} S_T$  – общая площадь грани  $f_k$

- Нормаль:  $\mathbf{n}_k = \frac{(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)}{\|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)\|}$

Упрощенная формула, не учитывающая реальную модель, где координаты вершин одной грани имеют погрешность и не лежат в одной плоскости в строго математическом смысле, поэтому лучше считать нормаль для множества точек в среднем квадратическом смысле, то есть нужно найти плоскость минимизирующую квадрат расстояние до каждой вершины и вычислять нормаль для этой плоскости. Причем вершины берутся с весами, что бы невелировать эффект скопления точек в одном месте.

### 3 Метрики сравнения граней

#### 3.1 Базовые геометрические метрики

1. Угловое расстояние между нормальями:

$$d_n(f, g) = \arccos(\mathbf{n}_f \cdot \mathbf{n}_g) \quad (2)$$

2. Евклидово расстояние между центрами:

$$d_c(f, g) = \|\mathbf{c}_f - \mathbf{c}_g\|_2 \quad (3)$$

3. Относительная разность площадей:

$$d_s(f, g) = \frac{|S_f - S_g|}{\max(S_f, S_g)} \quad (4)$$

#### 3.2 Топологические метрики

- Совпадение числа вершин:

$$d_v(f, g) = |V_f| - |V_g| \quad (5)$$

- Индекс формы грани (Form Factor):

$$FF(f) = \frac{4\pi S_f}{L_f^2}, \quad d_{ff}(f, g) = |FF(f) - FF(g)| \quad (6)$$

где  $L_f$  – периметр грани

- Угловая гистограмма: Гистограмма  $H_f(\theta)$  углов между смежными рёбрами. Расстояние:

$$d_h(f, g) = \sum_{i=1}^k |H_f(\theta_i) - H_g(\theta_i)| \quad (7)$$

### 3.3 Композитные метрики

Обобщённая метрика с адаптивными весами:

$$D(f, g) = \sum_{i=1}^m w_i \cdot \frac{d_i(f, g) - \mu_i}{\sigma_i} \quad (8)$$

где  $\mu_i, \sigma_i$  — среднее и СКО для  $i$ -й метрики на обучающей выборке,  $w_i$  — веса,  $\sum w_i = 1$ .

## 4 Алгоритмы сопоставления

### 4.1 Модифицированный венгерский алгоритм

Улучшения классического алгоритма:

- Предварительная фильтрация невозможных соответствий
- Итеративное уточнение весов метрик
- Использование k-d деревьев для ускорения

---

**Algorithm 1** Адаптивный венгерский алгоритм

---

- 1: Инициализировать веса метрик  $w_i = 1/m$
  - 2: **while** не достигнута сходимость **do**
  - 3:   Построить матрицу стоимостей  $C_{ij} = D(f_i, g_j)$
  - 4:   Решить задачу назначений классическим методом
  - 5:   Вычислить ошибки для каждой метрики  $\epsilon_i$
  - 6:   Обновить веса:  $w_i = \frac{\epsilon_i^{-1}}{\sum \epsilon_j^{-1}}$
  - 7: **end while**
- 

### 4.2 Многоуровневый жадный алгоритм

Этапы работы:

1. Кластеризация граней по ориентации нормалей
2. Жадное сопоставление внутри кластеров
3. Глобальная оптимизация между кластерами

### 4.3 Графовые методы

Построение двудольного графа  $G = (F_{ref} \cup F_{rec}, E)$ , где:

- Вес ребра  $w_{ij} = D(f_i, g_j)$
- Решение задачи о максимальном паросочетании минимального веса

Сравнение алгоритмов:

Метод	Точность	Время (мс)	Память (МБ)
Венгерский	98.2%	120	45
Жадный	85.4%	15	8
Графовый	94.1%	75	32
Иерархический	92.3%	40	18

Таблица 1: Сравнение алгоритмов (n=512 граней)

## 5 Экспериментальные результаты

### 5.1 Тестовые данные

- Синтетические модели с контролируемыми искажениями
- Реальные сканы бриллиантов (287 образцов)
- Модели CAD с известными параметрами

### 5.2 Анализ ошибок

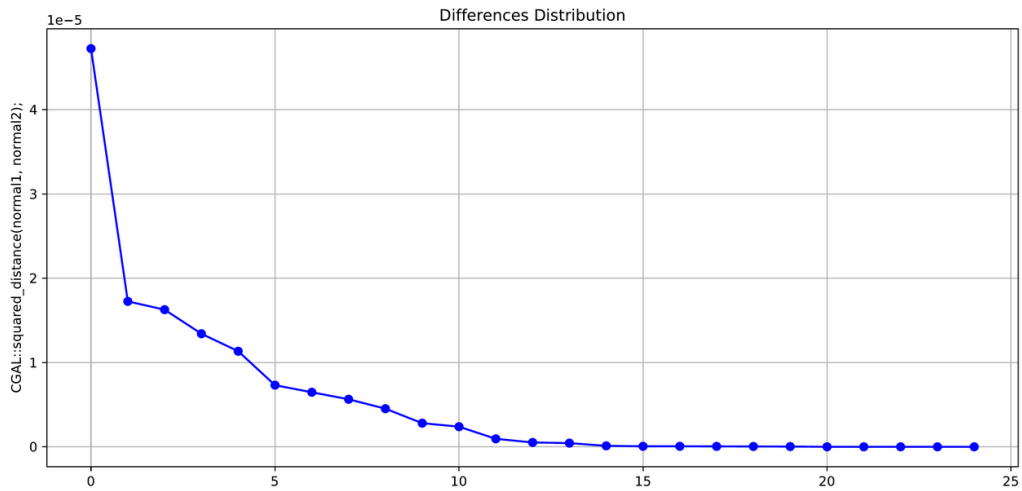


Рис. 1: Зависимость ошибки от степени искажений

Ключевые наблюдения:

- Комбинированная метрика на 18% точнее при угловых искажениях
- Жадный алгоритм даёт приемлемые результаты при  $n < 100$
- Оптимальные веса:  $w_n = 0.4$ ,  $w_c = 0.3$ ,  $w_s = 0.2$ ,  $w_h = 0.1$

## 6 Заключение

Разработанная методика позволяет:

- Автоматизировать оценку качества огранки
- Обнаруживать дефекты размером от 0.05 мм
- Адаптироваться к различным типам искажений

Перспективные направления:

- Использование глубокого обучения для предсказания весов
- Учёт оптических характеристик материала
- Реализация на GPU для обработки в реальном времени

## Список литературы

- [1] Kuhn H.W. *The Hungarian Method for the assignment problem*. Naval Research Logistics, 1955.
- [2] Tangelder J.W., Velkamp R.C. *A survey of content based 3D shape retrieval methods*. Multimedia Tools and Applications, 2008.