

ПРОГРАМА ДЛЯ АВТОМАТИЗАЦІЇ ВИКОНАННЯ РОБОЧИХ КРЕСЛЕНЬ ЗА ТРИВИМІРНОЮ МОДЕЛЛЮ ПОВЕРХНІ НЕСТАНДАРТНОГО ЗАСКЛЕННЯ БУДІВЛІ

У статті розглядається задача автоматизації побудови креслень фрагментованих складних поверхонь будівель та спосіб вирішення такої задачі з прикладами початкових, вихідних даних та алгоритмом роботи програми.

Опис проблеми

На сьогоднішній день в будівельній галузі все більшого значення набувають сміливі архітектурні та дизайнерські рішення щодо форми, конструкції та методів реалізації проектів споруд, особливо у містах, де значну увагу приділяють зовнішньому вигляду будівель та вулиць. В зв'язку з цим виникає питання розрахунку параметрів складних поверхонь, в даному випадку поверхні нестандартно заскленого фасаду будинку. У фахівців-архітекторів достатньо засобів для автоматизації побудови креслень для поверхонь (зі скла, або інших твердих листових матеріалів) побудованих з прямокутних та трикутних фрагментів. Але для поверхонь, складених із неправильних чотирикутників, відповідних програм не знайшлось.

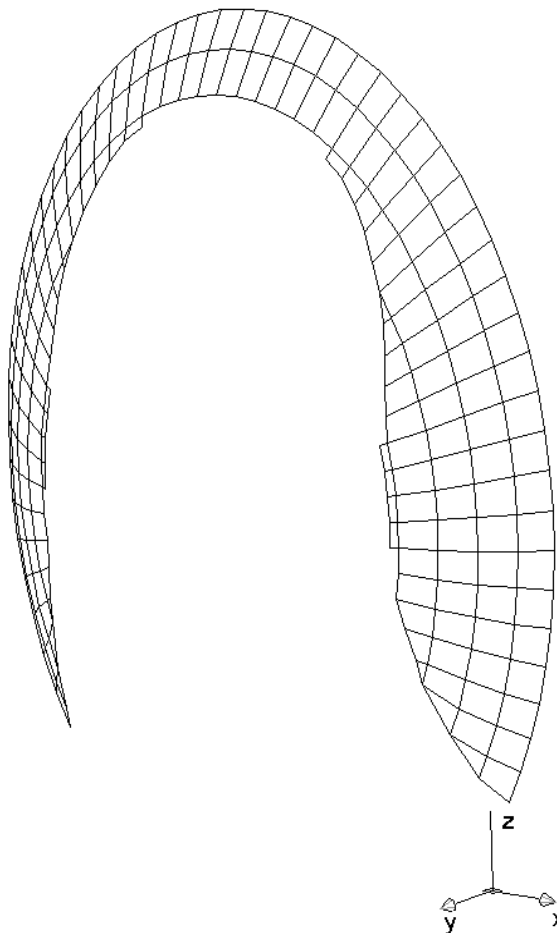
Постановка задачі

Архітектор-дизайнер пропонує креслення каркасу заскленої поверхні (можливе засклення фрагментами), попередньо розбитої на чотирикутники та трикутники (майбутні склопакети) у вигляді ліній в тривимірному просторі і координати перекриттів.

Необхідно автоматизувати виділення фрагментів з креслення каркасу та побудову робочих креслень окремих деталей та об'єкта в цілому.

Початкові дані

Тривимірна модель каркасу, побудована тільки лініями, подається у вигляді “.dxf”-файлу, де додатній напрямок осі Z відповідає додатному напрямку вертикалі.



Висоти перекриттів подаються у вигляді текстового файлу з рядком чисел.

Для розв'язку задачі намічено такі основні головні частини роботи.

1. Виявити за кресленням плоскі фрагменти поверхні три- або чотирикутної форми (враховуючи допуск відхилення однієї вершини з від площини – 3,0 мм), при виявленні помилки в кресленні програма має повідомити про це користувача. Зберегти координати вузлових точок (та приналежність цих точок відповідним фрагментам) у базу даних (текстові файли) з відповідними номерами (близько 40% загального часу роботи).

2. Розподілити вузлові точки між перекриттями. Зберегти в базу даних (текстовий файл) координати вузлових точок відносно найближчого нижчого перекриття, приймаючи центром координат точку (0, 0, висота_перекриття) (близько 5% загального часу роботи).

3. Побудувати 3-вимірну модель складної поверхні (або необхідні проекції, достатні для визначення положення всіх деталей) з нумерацією фрагментів (склопакетів) та вузлових точок каркасу (для ідентифікації вузлів базової металоконструкції), також умовно побудувати площини перекриттів (близько 30% загального часу роботи).

4. Побудувати плоске креслення всіх фрагментів, вказуючи необхідні розміри, та нумеруючи фрагменти відповідно до створеної бази даних. Помітити вершини фрагментів номерами вузлових точок, які належать відповідним вершинам фрагментів на тривимірному кресленні (при цьому необхідно врахувати, що на кресленні деталь має бути повернута зовнішньою стороною до користувача, а центр координат на кресленні каркасу знаходиться всередині будівлі) (близько 30% загального часу роботи).

При аналізі поставленої задачі зроблено такі зауваження.

1. При розпізнаванні “склопакетів” необхідно враховувати, що фрагмент обмежений трьома або чотирма лініями.

2. Допускається відхилення однієї з вершин чотирикутного фрагмента від площини інших трьох вершин на $\epsilon=3$ міліметри, інакше потрібно зафіксувати помилку та повідомити користувача.

3. Поверхня засklenня може бути не суцільна, а складатися з кількох відокремлених частин.

Для реалізації розв'язку задачі розроблено нижче описаний алгоритм.

1. Зчитування даних з DXF-файлу.

Відкрити DXF файл, циклічно знайти рядки з початковими і кінцевими координатами ліній, знести координати ліній у двовимірний масив.

2. Побудова графу.

3. Двовимірний масиву, виокремивши вершини та взаємозв'язки між ними, створити граф, де вершини графу – вершини каркасу, а взаємозв'язки графу – лінії каркасу.

3. Знаходження трикутників і чотирикутників.

Працюючи із графом виконати пошук в ширину для три-, та чотирикутників. При цьому потрібно врахувати, що чотирикутник, який складається з кількох трикутників, не є окремим фрагментом поверхні. Далі необхідно знайти, якими шляхами без повторів, можна із даної вершини повернутись в цю ж саму вершину через три або чотири кроки (бо одна вершина може належати до кількох три-, та чотирикутників, і всі ці фігури необхідно ідентифікувати);

Трикутники та чотирикутники занести до відповідних масивів.

Вивести їх кількість на екран.

4. Перевірка результатів.

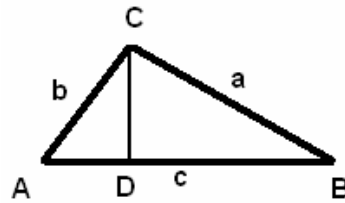
Оскільки у трикутника всі три точки завжди знаходяться на одній площині, то потрібно зробити перевірку тільки для чотири кутників на відхилення однієї з вершин від площини. Для цього потрібно вибрати три вершини чотирикутника, приймаючи їх за площину, та визначити аналітичну формулу для даної площини. Потім підставити четверту вершину, і якщо відхилення більше певного допуску ϵ , вивести попередження, що є помилка у початкових даних.

5. Занести дані про чотирикутники та трикутники до відповідних файлів, зчитати файл з координатами перекриттів та створити ще один файл, в який занести координати вершин каркасу відносно найближчого нижчого перекриття, згрупувавши їх за відповідними поверхнями.

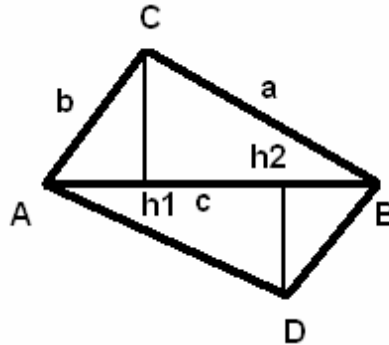
6. Потрібно кожний трикутник та чотирикутник у просторі перенести на фронтальну площину.

Для трикутників потрібно визначити сторону із найбільшою довжиною (c), а потім визначити довжини двох інших сторін і знайти висоту:

$$h = CD = \sqrt{b^2 - \left(\frac{b^2 - a^2 + c^2}{2c} \right)^2}$$



Для чотирикутника потрібно розбити його по діагоналі на два трикутники і побудувати спочатку верхній, а потім нижній, враховуючи положення вершин одна відносно одної.



Реалізація

Алгоритм реалізовано у вигляді програми, що на основі початкових даних у відповідному форматі формує текстові файли з необхідними даними та наочні тривимірні зображення у середовищі популярного інженерного програмного пакету.

Приклади результатів роботи програми.

1. База даних (текстовий файл) з номерами вузлових точок, та їх координатами. Приклад:

```
1 -1057,83911132812 11125,025390625 34064,627578125
2 -981,006225585936 12070,384765625 32829,772109375
3 -1955,01892089844 12070,384765625 32659,1773828125
4 -2108,13720703125 11125,025390625 33880,670546875
5 -2914,94555664062 12070,384765625 32376,2105859375
6 -3143,24584960937 11125,025390625 33575,545546875
7 -890,668029785155 13007,98046875 31377,8609765625
8 -1774,98645019531 13007,98046875 31222,975234375
9 -2646,51586914062 13007,98046875 30966,06703125
10 -3853,8701171875 12070,384765625 31982,9137109375
```

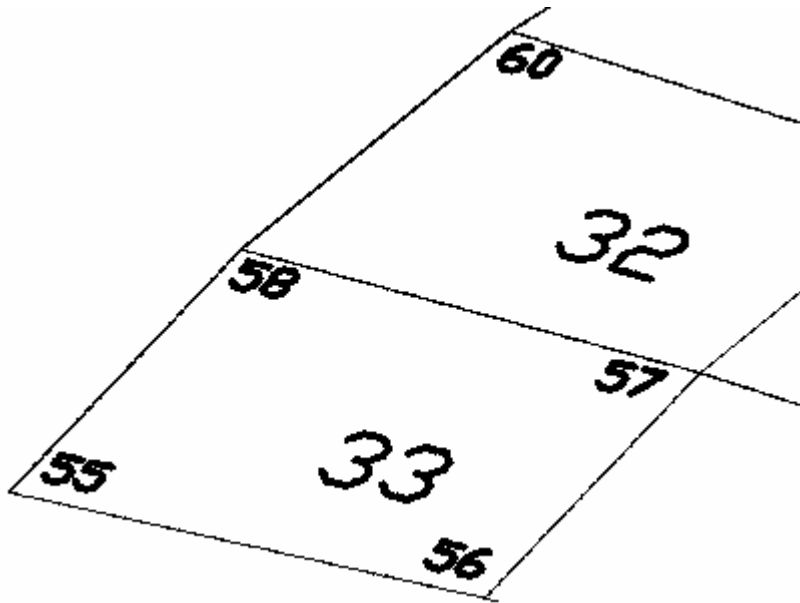
2. База даних (текстовий файл) з номерами фрагментів, та номерами відповідних їм вузлових точок. Приклад:

```
1:1 2 3 4
2:4 3 5 6
3:2 7 8
4:3 8 9 5
```

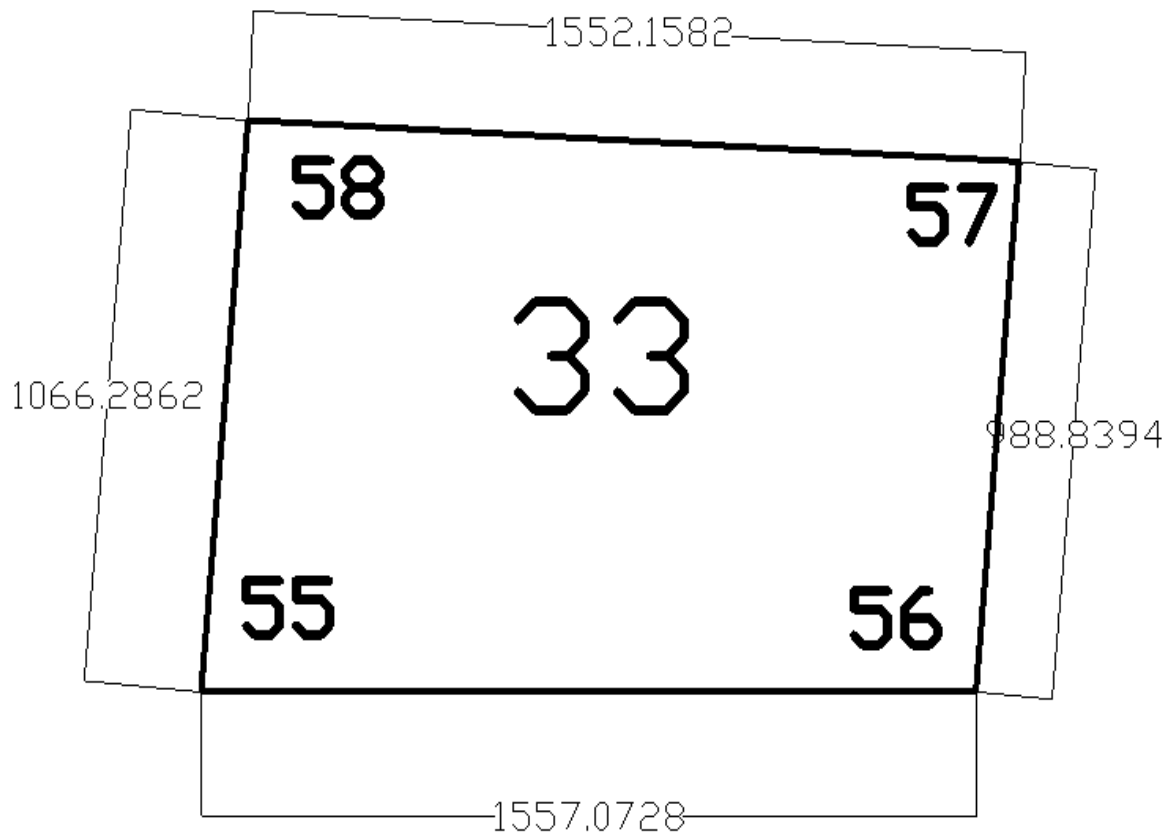
3. База даних (текстовий файл) з координатами вузлових точок відносно найближчого нижчого перекриття. Приклад:

```
Поверх 1
Поверх 2
67 : -11138,9 2358,76 2481,35
68 : -10329,86 3304,12 3039,29
69 : -9850,97 3304,12 1866,7
Поверх 3
57 : -20440,68 2358,76 2068,36
58 : -19554,96 3304,12 2329,46
59 : -19307,53 3304,12 1029,64
```

4. Тривимірна модель поверхні скляного покриття з умовним зображенням перекриттів та необхідні проекції. Кожен фрагмент та точка сходження пронумеровані. Приклад:



5. Плоске креслення фрагментів з вказуванням необхідних розмірів та номерами фрагментів і відповідних точок. Приклад:



Розроблена програма має реальне застосування у будівельній галузі. Крім того можливе використання поданого алгоритму при вирішенні подібних по суті інженерних задач.

Список використаних джерел

1. Оре О. Теория графов. 2-е изд. — М.: Наука, 1980. — 336 с.
2. Зыков А. А. Основы теории графов. — М.: «Вузовская книга», 2004. — 664 с.
3. Харари Ф. Теория графов. — М.: Мир, 1973. — 458 с.