

АНАЛІЗ ФАЙЛІВ ФОРМАТУ STL ЯК ВХІДНИХ ДАНИХ ДЛЯ СИСТЕМ ШВИДКОГО ПРОТОТИПУВАННЯ

Віталій ГУСЄВ¹, Тетяна НІКІФОРОВА²

^{1,2} Придніпровська державна академія будівництва та архітектури
24-а, вул. Чернишевського, Дніпро, Україна, 49005

¹husievvitalii@gmail.com, <https://orcid.org/0000-0001-6813-9824>

²nikiforova.tetiana@pdaba.edu.ua, <https://orcid.org/0000-0002-0688-2759>

Анотація. Моделювання будівельних конструкцій методом пошарового наплавлення (Fused Deposition Modeling – FDM) є одним із найпопулярніших методів адитивного виробництва, який використовує процедуру нашарування для створення 3D-моделей шар за шаром.

Оскільки шари є складовими тривимірної конструкції, сам процес нарізання шарів може безпосередньо впливати на обробку поверхні та механічні властивості конструкції, що є двома основними недоліками FDM. У зв'язку з цим, дослідження різних алго-ритмів нарізки може поглибити розуміння проблем і прогалин у FDM, що допоможе покращити якість 3D-друкованих будівельних об'єктів.

Алгоритми нарізки, що розроблені для технології FDM, поділяються на дві основні групи: планарні та непланарні. Необхідно проводити складні алгоритмічні обчислення, щоб дати змогу принтеру розгорнути багатоосьову площину та неплоскості, що є недоліком цих передових методів.

У статті розглянуті основні принципи слайсінгу та умовно окреслені можливі складнощі, що виникають під час реалізації технології FDM. Також детально описана структура файлу формату STL у бінарному та ASCII (англ. American standard code for information interchange) форматах.

Досліджені на сформовані загальні характеристики та обмеження формату файлів STL.

Наведений опис методів класичного слайсінгу моделі та механізми реалізації більш специфічних задач. Представлено актуальні виклики та перспективні розв'язки проблеми.

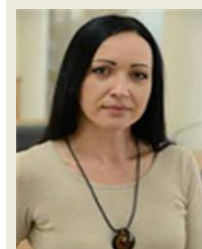
© В. ГУСЄВ, Т. НІКІФОРОВА, 2022

ментів і готових вузлів [1, 2]. Тривимірний друк використовує цифрові вхідні дані з



Віталій ГУСЄВ

аспірант кафедри залізобетонних і кам'яних конструкцій



Тетяна НІКІФОРОВА

декан будівельного факультету, д.т.н., професорка

Ключові слова. 3-D друк, STL-файли, САПР, 3D-моделювання, формат ASCII, метод пошарового наплавлення.

ВСТУП

Тривимірний друк – це високотехнологічний крок у розвитку швидкого прототипування у будівельній галузі.

Моделювання будівель і споруд, будівельних конструкцій та виробів методом пошарового наплавлення скорочує час і вартість проектування і полегшує пряме тане-пряме будівельне виробництво, створюючи фактичні будівельні об'єкти безпосередньо з цифрового введення, що робить можливим доставку найширшого доступного асортименту конструктивних еле-

ментів і готових вузлів [1, 2]. Тривимірний друк використовує цифрові вхідні дані з тривимірних даних і створює суцільні тривимірні об'єкти за допомогою адитивного

пошарового процесу. Простий у використанні, доступний за ціною та компактний для виробничого розгортання, 3-D друк широко використовується дизайнерами, інженерами для розробки концепцій і дизайну продукту, що прискорює процес проектування та скорочення часу спорудження об'єкту будівництва та введення його в експлуатацію. Останніми роками 3D-друк і 3D-принтери стали фінансово доступними для малого та середнього бізнесу, тим самим виводячи цей процес на масовий рівень виробництва та застосування у будівельній практиці. 3D-принтери також пропонують величезний потенціал для виробничих застосувань. Ця технологія знаходить широке застосування у реалізації та відтворенні складних будівельних моделей та конструкцій.

ПОСТАНОВКА ПРОБЛЕМИ ТА АНАЛІЗ ПОПЕРЕДНІХ ДОСЛІДЖЕНЬ

STL-файли, що створені системами 3D-моделювання, містять трикутне представлення поверхонь і стали стандартом для введення даних в технологіях швидкого прототипування.

У цих технологіях фізичні об'єкти створюються шар за шаром, кожен шар є двовимірним перерізом тривимірної сітки у форматі STL.

У сучасних системах САПР мозаїка поверхні часто закінчується помилками в структурі даних у вигляді прогалів і дірок, що призводить до відкритих петель у поперечних перерізах, які не можуть бути виготовлені як шари.

Однак з таким представленням у вигляді трикутних сіток, виникають дві великі проблеми [5, 6].

Одна з труднощів полягає в тому, що багато сучасних твердотільних моделей не створюють топологічно правильних мозаїк. Помилки, такі як прогалів та численні грані, призводять до неправильних поперечних перерізів, які не можуть бути виготовлені як шари. За останнє десятиліття багато дослідницьких груп працювали над спрощенням поверхні та опублікували багато робіт на цю тему.

Алгоритми спрощення сітки зменшують кількість трикутників у даній моделі, мінімально змінюючи її геометричну форму. Інша проблема полягає у високій складності таких сіток. Моделі з мільйонами трикутників часто вимагають більше ніж доступне сховище та тривалий час обробки. Тому, зменшення складності поверхневих сіток є обов'язковою вимогою для роботи з такими представленнями моделей.

Під час тесселяції в сітці дискретизації можуть виникнути численні проблеми з викривленням форми [3, 4]. Тому розробка дискретних алгоритмів для трикутних сіток з метою згладжування форми є ще однією центральною проблемою.

Такі алгоритми використовують результати та методи диференціальної геометрії, варіаційного дизайну поверхні та числового аналізу та знаходять застосування в багатьох областях.

Однак дуже мало робіт стосуються виявлення та виправлення топологічних помилок і побудови відповідних структур даних над трикутною сіткою, які також містять топологічну інформацію.

ВИКЛАДЕННЯ ОСНОВНОГО МАТЕРІАЛУ

STL – це формат файлу, який широко використовується для програмного забезпечення САПР у стерео літографії, створеного компанією 3D Systems Inc.

Цей формат файлу підтримується багатьма іншими програмними пакетами; він використовується для швидкого прототипування та автоматизованого виробництва. Файли STL описують лише геометрію поверхні тривимірної об'єкта без будь-якого представлення кольору, текстури чи інших загальних атрибутів моделі САПР. Формат STL визначає як ASCII, так і двійкове представлення. Двійкові файли є більш поширені, оскільки вони більш компактні [7].

Файл STL описує неструктуровану триангульовану поверхню за допомогою одиничної нормалі та вершин (упорядкованих за

правилом правої руки) трикутників із використанням тривимірної декартової системи координат.

Файл складається з набору трикутників, які можна апроксимувати з певною точністю для будь-якої поверхні. Поверхня задається набором трикутників, кожен з яких задається трьома вершинами і вектором нормалі, що однозначно задає положення трикутника в просторі.

Чим менші за розміром трикутники, тим краще з їх допомогою можна апроксимувати поверхню, відповідно зростає кількість трикутників в моделі [8, 9].

Інформація у STL файлах може бути записана в ASCII форматі або в бінарному форматі.

Кожен з цих форматів має свої переваги та недоліки. З файлом формату ASCII легше працювати, оскільки інформація в файлі зберігається у придатному до читання людині форматі.

В свою чергу, та ж інформація в бінарному форматі буде займати значно менше місця на дисковому просторі, але потребує конвертації з бінарного формату та знань про структуру файлу [10].

Сферична поверхня досить точно задана декількома тисячами трикутників, які містять достатньо інформації для відтворення зображення фігури на екрані (рис. 1).

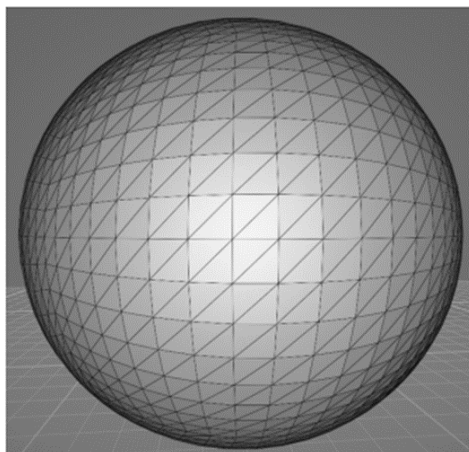


Рис. 1 Тривимірна модель в форматі STL

Fig. 1 Three-dimensional model in .STL format

Структура файлу STL в форматі ASCII дає можливість розібратися з форматом

файлу не шукаючи при цьому додаткової літератури.

```
solid Wall
  facet normal -1.000000e+000 0.000000e+000 0.000000e+000
    outer loop
      vertex 0.000000e+000 0.000000e+000 2.000000e+002
      vertex 0.000000e+000 1.000000e+002 2.000000e+002
      vertex 0.000000e+000 0.000000e+000 0.000000e+000
    endloop
  endfacet
  ...
endsolid
```

Рис. 2. Структура STL файлу в форматі ASCII

Fig. 2. The structure of the STL file in ASCII format

Кожен структурний елемент позначається окремими ключовим словом: solid – початок моделі, facet normal – вектор

нормалі трикутника (заданий трьома координатами), loop – початок циклу перебору вершин трикутника, vertex – вершина трикутника (задана трьома координатами), end

loop – кінець циклу перебору вершин трикутника, endfacet – кінець опису одного трикутника, endsolid – кінець опису моделі (рис. 2).

Бінарний формат має іншу структуру файлу (рис. 3). Перші 80 байт відводяться на зберігання назви моделі. Потім вказується кількість трикутників моделі (4 байти). Далі описується масив трикутників по 50 байт на

трикутник (12 байт на нормаль, по 12 байт на кожну вершину та 2 байти на зберігання кольору).

Кожен трикутник описується дванадцятьма 32-розрядними числами з плаваючою комою: три для нормалі, а потім три для координати X/Y/Z кожної вершини – як і у версії ASCII STL [11].

```

UINT8[80]      - Header
UINT32         - Number of triangles

for each triangle
  REAL32[3]    - Normal vector
  REAL32[3]    - Vertex 1
  REAL32[3]    - Vertex 2
  REAL32[3]    - Vertex 3
  UINT16       - Attribute byte count
end

```

Рис. 3 Структура .STL файлу у бінарному форматі
Fig. 3 The structure of the .STL file in ASCII format

Після дванадцятого числа з плаваючою точкою йде двобайтове «коротке» ціле число без знаку, яке є «підрахунком байтів атрибута» – у стандартному форматі це має бути нуль, оскільки більшість програм не розуміє нічого іншого.

Нормаль фасету має бути одиничним вектором, спрямованим назовні від суцільного об'єкта і в ASCII форматі, і в двійковій версії STL формату. У більшості програм це може бути встановлено на (0,0,0), і програмне забезпечення автоматично обчислить

нормаль на основі порядку вершин трикутника за допомогою «правила правої руки» [12, 13].

Також, слід забезпечити як нормальну грань, так і відповідний порядок вершин – навіть якщо це, здається зайвим. Деякі програми використовують нормаль для ефектів затінення, тому «нормалі», перелічені у файлі, не є нормальми справжніх граней. Слід дотримуватися правила від вершини до вершини (рис. 4).



Рис. 4 Правило від вершини до вершини
Fig. 4 The vertex-to-vertex rule

Кожен трикутник повинен мати дві вершини з кожним із сусідніх трикутників. Іншими словами, вершина одного трикутника не може лежати на стороні іншого.

Необхідно завжди дотримуватися загальних характеристик та враховувати обмеження формату файлів STL [15, 16].

Для 3D-принтерів, які можуть створити будь-яку об'ємну форму у вигляді серії зрізів, потрібна серія замкнутих двовимірних контурів, які заповнюються твердим матеріалом, оскільки шари зливаються разом.

Природним форматом файлу для такої машини буде серія замкнутих багатокутників, що відповідають різним Z-значенням. Однак, оскільки можна змінювати товщину шару для швидшої, але менш точної побудови, значно простіше визначити модель, яку потрібно побудувати, як замкнутий багатогранник, який можна розрізати на необхідних горизонтальних рівнях.

Можна стверджувати, що формат файлу STL, здатний визначити багатогранник із будь-якою багатокутною гранню але на практиці він використовується лише для трикутників, це означає, що значна частина синтаксису файлу є зайвою. Крім того, значення нормалі не потрібне, оскільки це є пряме обчислення координат трикутника з орієнтацією що контролюється за правилом правої руки.

Файли STL мають бути замкнутими та з'єднаними як комбінаторна поверхня, де кожне ребро трикутника є ребром двох суміжних трикутників. Оскільки синтаксис не передбачає застосування цієї властивості, її можна ігнорувати для програм, де близькість не має значення [14].

Близькість має значення лише в тому випадку, коли програмне забезпечення, яке розрізає трикутники, вимагає, щоб воно гарантувало замкнутість двовимірних полігонів. Іноді, таке програмне забезпечення можна написати для усунення невеликих розбіжностей шляхом переміщення кінцевих точок ребер, які знаходяться близько одна до одної, щоб вони збігалися. Результати непередбачувані, але часто цього достатньо, щоб отримати шуканий результат.

Очевидно, є багато можливостей для «вдосконалення» цього формату файлу, який у його нинішньому вигляді є нічим іншим, як переліком груп із 9 (або 12) чисел з плаваючою комою, вбудованих у якийсь непотрібний синтаксис. Оскільки кожна вершина в середньому буде використовуватися в шести різних трикутниках, значну економію пам'яті можна отримати, перерахувавши всі точки в таблиці на початку файлу, і закінчивши списком визначень трикутників, що складається з триплетів цілих чисел, які посилаються на цю таблицю. Однак для генерації одного контурного зрізу за допомогою дуже легкого програмного забезпечення на комп'ютері з невеликим об'ємом пам'яті цей формат є ідеальним, оскільки його можна обробити за один прохід, незалежно від розміру файлу [17].

Класичний підхід до слайсингу моделі полягає в пошуку ліній перетину між ріжучою площиною та площинами набору трикутників моделі. Площина обраного трикутника може перетинатися з ріжучою площиною одним з 5 способів: ріжуча площина перетинає ребро і проходить через вершину, проходить по ребру, проходить через одну вершину, перетинає два ребра трикутника або трикутник лежить на площині перетину [18].

Ріжуча площина підіймається вгору з рівним кроком доки не досягне найвищої точки моделі. Після кожної ітерації отримуємо один або декілька замкнутих контурів, що формуються з набору отриманих ліній перетину.

Для збільшення точності фізичної моделі заповнюємо пустоти в середині моделі обраним патерном (рис. 5). Сітка патерну заповнює все поле для друку, а межі обраної моделі обрізають патерн, залишаючи лише частину сітки, що знаходиться всередині моделі.

Наступним кроком є генерація шляху руху екструдера для друку моделі матеріалом для друку.

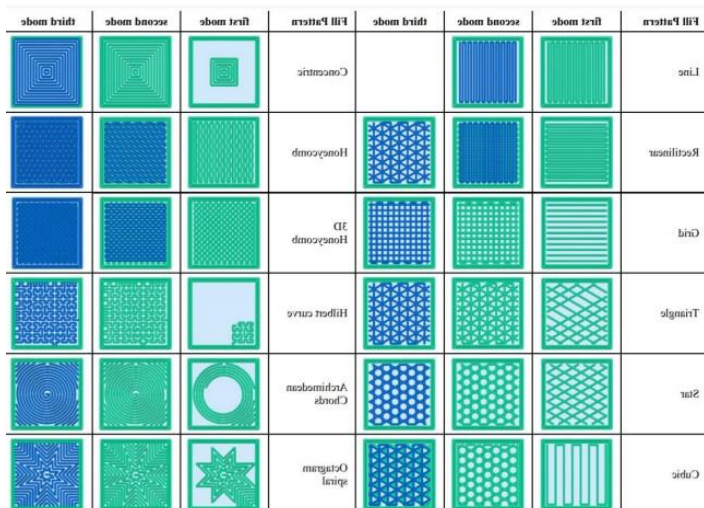


Рис. 5 Набір стандартних патернів заповнення
Fig. 5 A set of standard fill patterns

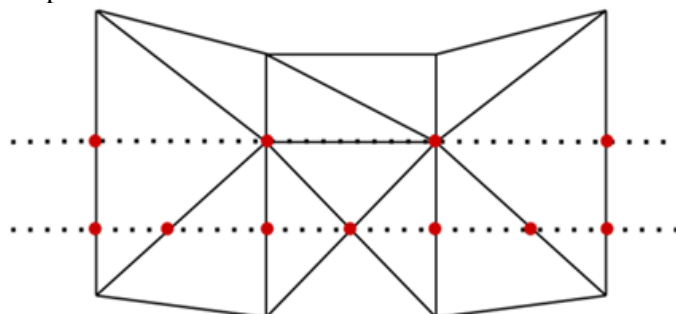


Рис. 6. Перетин ріжучої площини з трикутниками моделі
Fig. 6. Intersection of the cutting plane with the triangles of the model

На рис. 6 пунктиром зображено ріжучі площини, а червоним позначено точку перетину цих площин з трикутними гранями моделі.

Пунктирні лінії між червоними точками і є лініями контурів для подальшого їх заповнення.

В дослідженні впливу обраного патерну на механічні властивості надрукованих елементів було виявлено, що обраний патерн та відсоток заповнення впливає на механічні властивості фігури. Те, як лінії патерну перетинаються один з одним і забезпечує міцність.

Спочатку друкуються контури фігури, потім вже заповнюється порожнини між стінками фігури [19]. Патерн, що однотипно повторюється просто обрізається стінками моделі. Таким чином, отримуємо заповнення лише всередині моделі. Такий спосіб генерації заповнення є рентабельним при друці пластиком і забезпечує можливість

використання однотипних патернів на різних моделях. Досить широкий набір патернів дає можливість вибрати потрібний патерн в залежності від задач.

Існує також швидший алгоритм нарізання мешу (меш-об'єкти Blender – це тривимірні геометричні примітиви, на основі яких створюють більш складні зображення) на шари.

Суть покращеного алгоритму нарізання полягає в тому, що період нарізання залежить від швидкості зміни моделі.

Тобто, якщо модель не змінюється в висоту, то використовуємо дані попереднього шару, а якщо змінюється інтенсивно, то нарізаємо модель частіше [20, 21]. Такий підхід дозволяє точніше відтворити фігуру, якщо 3D принтер дозволяє надрукувати шар потрібним розміром. Таким чином отримуємо більшу точність там, де модель інтенсивно змінюється. Але, якщо модель змінюється плавно, то накопчується незначне відхилення від реальних розмірів моделі.

Існують також методи слайсингу для більш специфічних задач. Наприклад, існує метод сферичного слайсингу, який ідеально підходить для слайсингу сфероподібних об'єктів. Метод передбачає, що нарізання моделі відбувається сферичними площинами різного радіусу починаючи від центра моделі. Відтворити згенерований набір інструкцій зможе лише принтер на основі робота-маніпулятора.

ВИСНОВКИ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

На даний час існує необхідність дослідження детального огляду методів дискретного моделювання поверхні, які необхідні для швидкого створення прототипів, таких як керування файлами STL, аналіз трикутної сітки у форматі STL для виявлення та усунення топологічних дефектів, побудова відповідних структур даних для опису форми і згладжування сіток.

На сьогодні виконаний аналіз літературних джерел, які зосереджені на геометрії цих проблем, і які представляють математичні інструменти. Для подальших досліджень необхідно дослідити на практичному рівні прийоми керування трикутними сітками. У процесі трансформації моделі STL може бути багато помилок даних, таких як отвір, тріщина, відсутня позиція, зворотний нормальний вектор, надмірність тощо, що пояснюється наявністю помилки опції даних і можливістю відсутності трикутника.

Це може спричинити відхилення у наступному швидкому прототипуванні та вплинути на застосування моделей STL.

Необхідно отримати на практичному досліді відношення точок, ребер і поверхонь шляхом створення топології для моделі STL, а також математично описати структуру даних графа, який представляє ребра в одному трикутнику та на основі різних характеристик помилок розробити алгоритми їх усунення.

Помилки отворів можна виправити шляхом додавання трикутників, які створюються точками, записаними в алгоритм пе-

ревірки отворів на основі принципу мінімального кута, тріщини можна виправити, змінивши координату точки. За допомогою різних додатків на основі мови програмування C++ (як приклад, Microsoft Visual C++) розробити систему перевірки помилок.

ЛІТЕРАТУРА

1. **3d Printing** Trends in Building and Construction Industry / P. Mehar et al. *International Journal of Scientific Research in Science and Technology*. 2020. P. 314–319.
2. **A critical review** of 3D printing and digital manufacturing in construction engineering / M. H. Ali et al. *Rapid Prototyping Journal*. 2022.
3. **El-Sayegh S., Romdhane L., Manjikian S.** A critical review of 3D printing in construction: benefits, challenges, and risks. *Archives of Civil and Mechanical Engineering*. 2020. Vol. 20, no. 2.
4. **Galantucci L. M., Percoco G., Dal Maso U.** A volumetric approach for STL generation from 3D scanned products. *Journal of Materials Processing Technology*. 2008. Vol. 204, no. 1-3. P. 403–411.
5. **Huang L. L., Zhang X. W.** Automatic Identification of Features from CAD Models Based on STL Files. *Applied Mechanics and Materials*. 2012. Vol. 220-223. P. 2524–2527.
6. **Huang S. H., Zhang L. C., Han M.** An Effective Error-Tolerance Slicing Algorithm for STL Files. *The International Journal of Advanced Manufacturing Technology*. 2002. Vol. 20, no. 5. P. 363–367.
7. **Koc B., Ma Y., Lee Y.** Smoothing STL files by Max-Fit biarc curves for rapid prototyping. *Rapid Prototyping Journal*. 2000. Vol. 6, no. 3. P. 186–205.
8. **Kumar A. V.** A review paper on 3d-printing and various processes used in the 3d-printing. *International journal of scientific research in engineering and management*. 2022. Vol. 06, no. 05.
9. **Leong K. F., Chua C. K., Ng Y. M.** A study of stereolithography file errors and repair. Part 1. Generic solution. *The International Journal of Advanced Manufacturing Technology*. 1996. Vol. 12, no. 6. P. 407–414.
10. **Leong K. F., Chua C. K., Ng Y. M.** A study of stereolithography file errors and repair. Part 2. Special cases. *The International Journal of Advanced Manufacturing Technology*. 1996. Vol. 12, no. 6. P. 415–422.

11. **Life cycle assessment of a concrete** 3D printing process / C. Roux et al. *The International Journal of Life Cycle Assessment*. 2022.
 12. **Nayyeri P., Zareinia K., Bougherara H.** Planar and nonplanar slicing algorithms for fused deposition modeling technology: a critical review. *The International Journal of Advanced Manufacturing Technology*. 2022. Vol. 119, no. 5-6. P. 2785–2810.
 13. **Nebrida J. A.** Automated Onsite Construction: 3D Printing Technology. *Journal of Engineering Research and Reports*. 2022. P. 47–55.
 14. **Robotics technologies** aided for 3D printing in construction: a review / Z. Xu et al. *The International Journal of Advanced Manufacturing Technology*. 2021. Vol. 118, no. 11-12. P. 3559–3574.
 15. **STL file generation** from measured point data by segmentation and Delaunay triangulation / S.-H. Lee et al. *Computer-Aided Design*. 2002. Vol. 34, no. 10. P. 691–704.
 16. **Szilvsi-Nagy M., Matyas G.** Analysis of STL files. *Mathematical and Computer Modelling*. 2003. Vol. 38, no. 7-9. P. 945–960.
 17. **Taufik M., Jain P. K.** On the achieving uniform finishing allowance through identifying shape deviation for additive manufacturing. *Australian Journal of Mechanical Engineering*. 2021. P. 1–19.
 18. **Umaras E., Tsuzuki M. S. G.** Additive Manufacturing - Considerations on Geometric Accuracy and Factors of Influence. *IFAC-PapersOnLine*. 2017. Vol. 50, no. 1. P. 14940–14945.
 19. **Yu W., Nie Z., Lin Y.** Research on the slicing method with equal thickness and low redundancy based on STL files. *Journal of the Chinese Institute of Engineers*. 2021. Vol. 44, no. 5. P. 469–477.
 20. **Zhang Z., Joshi S.** An improved slicing algorithm with efficient contour construction using STL files. *The international journal of advanced manufacturing technology*. 2015. Vol. 80, no. 5-8. P. 1347–1362.
 21. **Гусєв В. О., Нікіфорова Т. Д.** Дослідження етапів перетворення об'ємної моделі виробу в керуючий код для 3D-принтера в контексті автоматизованого будівництва технології 3D – друку. *Український журнал будівництва та архітектури*. – 2022. – №4. – С. 38-45. DOI:10.30838/J.BPSACEA.2312.250822.38.87
2. **A critical review of 3D printing** and digital manufacturing in construction engineering / M. H. Ali et al. *Rapid Prototyping Journal*. 2022.
 3. **El-Sayegh S., Romdhane L., Manjikian S.** A critical review of 3D printing in construction: benefits, challenges, and risks. *Archives of Civil and Mechanical Engineering*. 2020. Vol. 20, no. 2.
 4. **Galantucci L. M., Percoco G., Dal Maso U.** A volumetric approach for STL generation from 3D scanned products. *Journal of Materials Processing Technology*. 2008. Vol. 204, no. 1-3. P. 403–411.
 5. **Huang L. L., Zhang X. W.** Automatic Identification of Features from CAD Models Based on STL Files. *Applied Mechanics and Materials*. 2012. Vol. 220-223. P. 2524–2527.
 6. **Huang S. H., Zhang L. C., Han M.** An Effective Error-Tolerance Slicing Algorithm for STL Files. *The International Journal of Advanced Manufacturing Technology*. 2002. Vol. 20, no. 5. P. 363–367.
 7. **Koc B., Ma Y., Lee Y.** Smoothing STL files by Max-Fit biarc curves for rapid prototyping. *Rapid Prototyping Journal*. 2000. Vol. 6, no. 3. P. 186–205.
 8. **Kumar A. V.** A review paper on 3d-printing and various processes used in the 3d-printing. *International journal of scientific research in engineering and management*. 2022. Vol. 06, no. 05.
 9. **Leong K. F., Chua C. K., Ng Y. M.** A study of stereolithography file errors and repair. Part 1. Generic solution. *The International Journal of Advanced Manufacturing Technology*. 1996. Vol. 12, no. 6. P. 407–414.
 10. **Leong K. F., Chua C. K., Ng Y. M.** A study of stereolithography file errors and repair. Part 2. Special cases. *The International Journal of Advanced Manufacturing Technology*. 1996. Vol. 12, no. 6. P. 415–422.
 11. **Life cycle assessment of a concrete** 3D printing process / C. Roux et al. *The International Journal of Life Cycle Assessment*. 2022.
 12. **Nayyeri P., Zareinia K., Bougherara H.** Planar and nonplanar slicing algorithms for fused deposition modeling technology: a critical review. *The International Journal of Advanced Manufacturing Technology*. 2022. Vol. 119, no. 5-6. P. 2785–2810.
 13. **Nebrida J. A.** Automated Onsite Construction: 3D Printing Technology. *Journal of Engineering*

REFERENCES

1. **3d Printing Trends** in Building and

- Research and Reports*. 2022. P. 47–55.
14. **Robotics technologies aided** for 3D printing in construction: a review / Z. Xu et al. *The International Journal of Advanced Manufacturing Technology*. 2021. Vol. 118, no. 11-12. P. 3559–3574.
 15. **STL file generation** from measured point data by segmentation and Delaunay triangulation / S.-H. Lee et al. *Computer-Aided Design*. 2002. Vol. 34, no. 10. P. 691–704.
 16. **Szilvsi-Nagy M., Mátyási G.** Analysis of STL files. *Mathematical and Computer Modelling*. 2003. Vol. 38, no. 7-9. P. 945–960.
 17. **Taufik M., Jain P. K.** On the achieving uniform finishing allowance through identifying shape deviation for additive manufacturing. *Australian Journal of Mechanical Engineering*. 2021. P. 1–19.
 18. **Umaras E., Tsuzuki M. S. G.** Additive Manufacturing - Considerations on Geometric Accuracy and Factors of Influence. *IFAC-PapersOnLine*. 2017. Vol. 50, no. 1. P. 14940–14945.
 19. **Yu W., Nie Z., Lin Y.** Research on the slicing method with equal thickness and low redundancy based on STL files. *Journal of the Chinese Institute of Engineers*. 2021. Vol. 44, no. 5. P. 469–477.
 20. **Zhang Z., Joshi S.** An improved slicing algorithm with efficient contour construction using STL files. *The international journal of advanced manufacturing technology*. 2015. Vol. 80, no. 5-8. P. 1347–1362.
 21. **Husiev V. O., Nikiforova T. D.** Doslidzhennia etapiv peretvorennia ob'iemnoi modeli vyrobu v ke-ruiuchy kod dlia 3D-pryntera v konteksti avtoma-tyzovanoho budivnytstva tekhnolohii 3D – druku. *Ukrainskyi zhurnal budivnytstva ta arkhitektury*. – 2022. – №4. – S. 38-45.
DOI:10.30838/J.BPSACEA.2312.250822.38.876

ANALYSIS OF STL FILES AS INPUT DATA FOR RAPID PROTOTYPING SYSTEMS

*Vitalii HUSIEV
Tetiana NIKIFOROVA*

Summary. Since layers are components of a three-dimensional structure, the very process of cutting layers can directly affect the surface finish and mechanical properties of the structure, which are the two main disadvantages of FDM.

In this regard, the study of different slicing algorithms can deepen the understanding of problems and gaps in FDM, which will help improve the quality of 3D printed building objects.

Cutting algorithms developed for FDM technology are divided into two main groups: planar and non-planar. Complex algorithmic calculations are required to enable the printer to deploy multi-axis plane and non-plane, which is a drawback of these advanced methods.

The article discusses the basic principles of slicing and conditionally outlines the possible difficulties that arise during the implementation of FDM technology, and also describes in detail the structure of the STL format file in binary and ASCII (English American standard code for information interchange) formats.

The general characteristics and limitations of the STL file format were studied.

The description of methods of classic slicing of the model and the mechanisms of implementation of more specific tasks are given.

Current challenges and promising solutions to the problem are presented.

Keywords. Building objects, 3D printing, STL files, CAD, 3D modeling, ASCII format, fused deposition modeling

Стаття надійшла до редакції 12.11.2022 р.