

Висновки

Вибір способу розв'язання задачі типізації або встановлення еквівалентності схем залежить від обраної моделі представлення схем, а також наявних ресурсів і бажаної точності отриманих результатів. Потрібно також зауважити, що описані підходи можна застосовувати для встановлення ізоморфізму графів [4] або виділення в графі (пошук у графі) ізоморфних підграфів.

1. Bilal Radi A'Ggel Al-Zabi, Andriy Kernytskyu, Sergiy Tkatchenko. *Ways of Determination of Equivalence of Schemes. – Proceedings of the Xth International Conference "The Experience of Design and Application of CAD systems in Microelectronics" // IEEE CADSM 2009, Polyana-Svalyava, 2009. – P.218-219.* 2 Bilal Radi A'Ggel Al-Zabi, Kernytskyu A., Tkatchenko S. *Solution of Typization Task // IEEE MEMSTECH'2009. – Polyana, 2009. – p.84.* 3. Білаль Аль-Забі, Керницький А.Б., Ткаченко С.П. *Встановлення існування необхідних умов еквівалентності схем // Вісник Нац. ун-ту "Львівська політехніка", №591. Комп'ютерні системи проектування. Теорія і практика. – Львів, 2007. – С.87–90.* 4. А'Ггель Б.Р. Аль-Забі, Керницький А.Б., Ткаченко С.П. *Нові інваріанти встановлення ізоморфізму графів. // Вісник Нац. ун-ту "Львівська політехніка", №626. Комп'ютерні системи проектування. Теорія і практика. – Львів, 2008. – С.90–93.* 5. Базилевич Р.П., Ткаченко С.П. *Решение задачи разбиения методом паралельного свертывания // Вычислительная техника, VII. – Каунас: КПИ, 1975. – С.295–298.*

УДК 004.415.2

В.В. Сегеда

Національний університет "Львівська політехніка",
кафедра систем автоматизованого проектування

АЛГОРИТМИ ФОРМУВАННЯ ОПТИМАЛЬНОЇ ТРАЄКТОРІЇ РУХУ РОБОЧОГО ІНСТРУМЕНТУ У ВЕРСТАТАХ З ЧПК

© Сегеда В. В., 2009

Описано і проаналізовано алгоритми, які використовуються під час формування оптимальної траєкторії руху інструменту у верстатах з ЧПК. Описано модифікований алгоритм мінімальної відстані для створення траєкторії руху інструменту у верстатах з ЧПК.

Ключові слова – траєкторія руху, алгоритм, мінімальна відстань, верстат ЧПК

Described and analysed algorithms which use for forming optimum tool motion trajectory in CNC machines. Described modified minimum distance algorithm for forming optimum tool motion trajectory in CNC machines.

Keywords – motion trajectory, algorithm, minimum distance, CNC machine

Актуальність досліджень

Сьогодні, у зв'язку з швидким розвитком технологій, все частіше верстати з ЧПК використовують у різних сферах людської діяльності. Наприклад, у процесі нанесення графічних зображень на матеріал. Верстати з ЧПК дозволяють замінити людину і повністю автоматизувати процес гравірування зображень. Для виконання роботи верстатом виникає завдання формування траєкторії руху різального інструменту. Для цього сьогодні відомо декілька алгоритмів, кожен з яких має свої переваги і недоліки. Тому є актуальним детальний опис і порівняльна характеристика вже існуючих алгоритмів, створення нових оптимізуючих алгоритмів.

Постановка задачі

Задачі, які виникають у процесі створення програми керування механізмом переміщення різального інструменту (МПРІ) верстата з ЧПК :

- Оптимізація часу виконання операцій.
- Аналіз зображення і оптимізація кількості елементів для обробки.
- Визначення оптимального розміру заготовки для гравірування заданого зображення.

Оскільки існуючі програми керування МПРІ верстата, після налаштувань, самі керують процесом пришвидшення і сповільнення руху, залишається лише задати траєкторію, за якою має відбуватись цей рух і отримаємо результат. Тому залишається розв'язати задачу формування оптимальної траєкторії руху різального інструменту за затратами часу на гравірування зображення.

$$T = L_{\Sigma} / (s_z * z * n) \rightarrow \min, \quad (1)$$

де T – час затрачений на гравірування зображення, L_{Σ} – сумарна довжина переміщень інструменту, мм; s_z – подача інструменту на один різець, мм/зуб; z – число різців; n – частота обертання шпинделя, об/хв.

Аналіз існуючих алгоритмів формування траєкторії руху різального інструменту у верстатах з ЧПК при точковому гравіруванні зображень:

1. Рух по одній з осей з обробкою всіх необхідних елементів (зигзагоподібний рух).

На основі аналізу зображення, відомого розміру заготовки, на котрій воно має бути зображене, відомого розміру різального інструменту (для прикладу фреза діаметром 1 мм) визначається крок Δk руху інструменту по кожній з осей і точки, які буде оброблено. Траєкторія руху різального інструменту будується за алгоритмом: інструмент рухається, для прикладу, по осі X від 0 до i -ї координати і обробляє всі необхідні точки. Координата Y збільшується на крок Δk і рух по осі X відбувається від i -ї координати до 0 з обробкою всіх необхідних точок. Далі все повторюється аж доки не будуть оброблені всі точки, які утворюють зображення. Тобто по всій площині заготовки, на котрій має бути розміщений кінцевий малюнок, проходить інструмент. При формуванні траєкторії створюється G-код, в ньому для обробки однієї точки потрібно записати три коди. Один для переміщення в точку, яку потрібно обробити, і ще два – для опускання і піднімання інструменту.

Приклад записування в файлі G-коду для гравірування однієї точки:

```
G00 X5 Y4,5  
G00 Z0  
G00 Z3,5
```

Недолік цього алгоритму: тривалий час виконання операції, оскільки відстань між точками, які потрібно нанести на заготовку, може бути великою і інструмент проходить її в повітрі, незважаючи на те, що поряд можуть бути інші точки (але, щоб їх обробити потрібне зміщення по іншій осі). На рис.1 показано приклад руху інструменту за цим алгоритмом. За мету ми маємо вигравірувати 5 ліній. Інструмент обробляє точки 1–5, траєкторія його руху збігається з лінією 6 (в конкретний момент часу). Зрозуміло, що інструмент багато відстані проходить “без роботи”. Для прикладу між точками 1 і 2, хоча міг би обробити спочатку точки, які лежать біля точки 1, далі точку 2 і ті точки, які є біля неї і т. д.

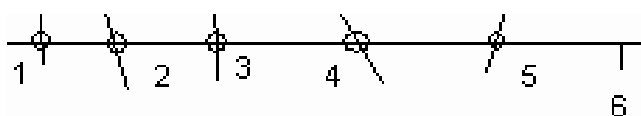


Рис. 1. Приклад траєкторії руху інструменту по одній з осей

2. Розбивання зображення на певну кількість згрупованих областей і обробка по черзі кожної з них.

Як і в попередньому випадку, після обробки зображення і всіх відомих величин (розмір заготовки і фрези), ми отримуємо набір тих точок, які потрібно нанести на заготовку. Алгоритм використовує евклідову відстань, яку обчислюємо за формулою [1, 2]:

$$D = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}, \quad (2)$$

де (x_1, y_1) і (x_2, y_2) – координати точок, відстань між якими обчислюється.

Вибирається певна початкова точка X_1 , яказначається центром першої області Z_1 . Найчастіше це береться точка з координатами $(0,0)$. Алгоритм, передусім виявляє найвіддаленіші елементи для обробки. Відшукується елемент, розміщений від X_1 найдалі, який призначається центром області Z_2 . Це здійснюється знаходженням відстані від точки X_1 до всіх інших точок, і знаходженням серед цих відстаней найбільшої. На третьому кроці алгоритму здійснюється обчислення відстаней між всіма іншими елементами обробки і центрами Z_1 , і Z_2 . В кожній парі цих вибірок вибирається мінімальне. Після цього виділяється максимальне з цих мінімальних відстаней. Якщо останнє становить значну частину відстані між кластерами Z_1 і Z_2 (половина цієї відстані або якась інша наперед визначена величина), відповідний елемент призначається центром області Z_3 . Інакше – виконання алгоритму припиняється. У загальному випадку описана процедура повторюється до тих пір, поки на якомусь кроці не буде отримане максимальне значення відстані, для якої умова, що викликає створення області, не виконується.

Отже, отримуємо точки, які потребують обробки, згруповані по групах, обробка здійснюється обробкою всіх елементів групи по зигзагоподібній траєкторії (описана вище) і після цього здійснюється перехід до елементів наступної групи. Коли елементи груп відомі, формується G-код для обробки кожного елемента кожної групи по зигзагоподібному алгоритму. Відповідно, елементи будуть оброблятися по групах (рис.2).

Недолік цього алгоритму: якщо в виділених областях (великого зображення) елементи для обробки будуть віддалені на певну відстань, то інструмент пройдёт цю відстань “без роботи” (при обробці по зигзагоподібному алгоритму). Відповідно будуть неоптимальні витрати часу. Приклад такої ситуації можна побачити на рис. 2. Для прикладу область 4 складається з віддалених елементів. Крім того, ще існує проблема мінімальних по часу переходів між областями. Потрібно аналізувати кожну область і в порядку їх обробки розміщати поряд ті області, відстань між елементами (які оброблятимуться першими або останніми) яких є мінімальна.

3. Рух інструменту по сусідніх елементах обробки на основі мінімальної відстані.

Після аналізу і обробки зображення, і всіх відомих величин (розмір матеріалу і діаметр фрези), ми отримуємо набір тих точок, які потрібно обробити. Алгоритм використовує евклідову відстань, яка обчислюється за формулою (2). При формуванні траєкторії (створення G-коду), починаючи з певного заданого початкового положення різального інструменту знаходиться точка, яка найближче розміщена до положення інструменту за формулою (2), методом знаходження відстаней від даної точки до інших точок, які потребують обробки. Серед цих відстаней знаходиться мінімальна, вона і визначатиме точку, яка оброблятиметься наступною. Код переміщення в цю точку і код її обробки записується в файл G-коду. Інформація про дану точку на наступних етапах алгоритму не використовується. Далі знаходиться наступна точка, яка є найближче до положення інструменту за принципом, описаним вище. Такий пошук і запис у файл траєкторії відбувається доки є точки для обробки. Різальний інструмент рухається з мінімальними переміщеннями “в повітрі”.

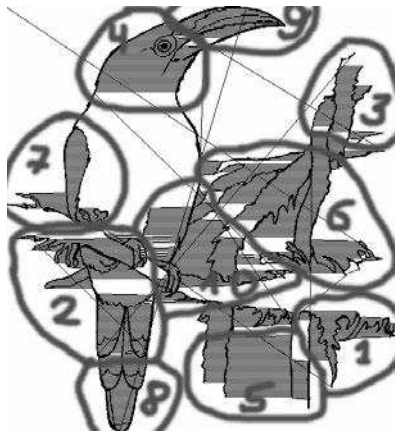


Рис. 2. Приклад роботи алгоритму поділу зображення на певну кількість частин



Рис. 3. Візуальне представлення неоптимальних переміщень інструменту при роботі по траєкторії, яка отримана на основі обчислення мінімальної відстані між елементами обробки

Недолік даного алгоритму: часто виникає ситуація, коли інструмент заходить в “глухий кут”, коли на прийнятній відстані біля нього немає елемента для обробки і для того, щоб дійти до цього елемента інструмент переміщається “без роботи” на велику відстань (більшу в десять і більше разів від середньої величини його переміщень). Щоб мінімізувати виникнення даної ситуації і відстані, які потрібно проходити, потрібно передбачати і аналізувати траєкторію руху наперед, що ускладнить даний алгоритм, але дозволить оптимізувати час роботи верстата. Бачимо необхідність модифікації алгоритму для даної задачі.

Перевага цього алгоритму: Серед наведених вище трьох алгоритмів, траєкторія руху інструменту, яка створена на основі даного алгоритму буде найоптимальнішою по часу роботи верстата.

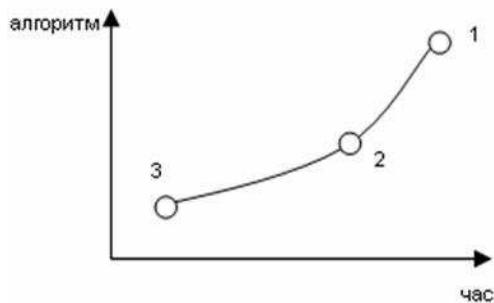


Рис. 4. Схематичне зображення зміни часу обробки залежно від того алгоритму, який використовується для формування траєкторії руху різального інструменту



Рис. 5. Схематичне зображення співвідношення проценту заповнення зображення і неоптимальної відстані, яку проходить інструмент (по траєкторії, створеній на основі мінімальної відстані) і яку можна оптимізувати

Модифікований алгоритм мінімальної відстані

Як було описано вище, алгоритм формування траєкторії руху різального інструменту на основі мінімальної відстані має недоліки, ці недоліки виправляє його модифікований варіант, який тут пропонується. Перший крок модифікованого алгоритму мінімальної відстані повністю збігається з алгоритмом мінімальної відстані. В результаті отримуємо сукупність точок (які потрібно обробити) і відому послідовність їх обробки з точки зору мінімальної віддалі. Наступним завданням є знаходження невеликих фрагментів зображення, для обробки яких різальний інструмент переміщується на велику відстань T_b (рис. 6,б). Потрібно визначити поняття T_b . Для цього знаходимо середню відстань T_e , яку проходить інструмент між елементами, які будуть оброблятися. Для

прикладу, поняття T_b може визначатись як $T_b = T_e * k$, де k – задана константа. Знаючи величину T_b відбувається послідовне обчислення відстаней між точками, які будуть обробляться. Кожна знайдена відстань T_i (i – загальна кількість точок для обробки) порівнюється з T_b . Якщо $T_i > T_b$, то цю точку записуємо в окрему послідовність. Перебравши, отже, всі точки і знайшовши точки, відстань до яких більша, ніж прийнятна T_b , ми отримуємо набір точок, час вирізання яких потрібно зменшити. Далі знаходимо точки, які знаходяться біля щойно знайдених на відстані, меншій від T_e . Це точки, які разом утворюють групу і які треба обробляти послідовно. Знайти ці точки можна з послідовності, яка отримана на кроці 1. Виключаємо їх з початкового набору точок для обробки. У початковій послідовності точок для обробки (отримана на кроці 1), яка тепер є модифікованою послідовністю, знаходимо точки, які є найближче до щойно знайдених груп елементів для обробки і вставляємо ці групи елементів після них в модифікованій послідовності. Проходячи послідовно модифіковану послідовність, формуємо G-код для точкового гравірування графічного зображення. В результаті оптимізували час вирізання зображення методом зменшення загального шляху (за однакової якості обробки), який проходить різальний інструмент (рис. 6).

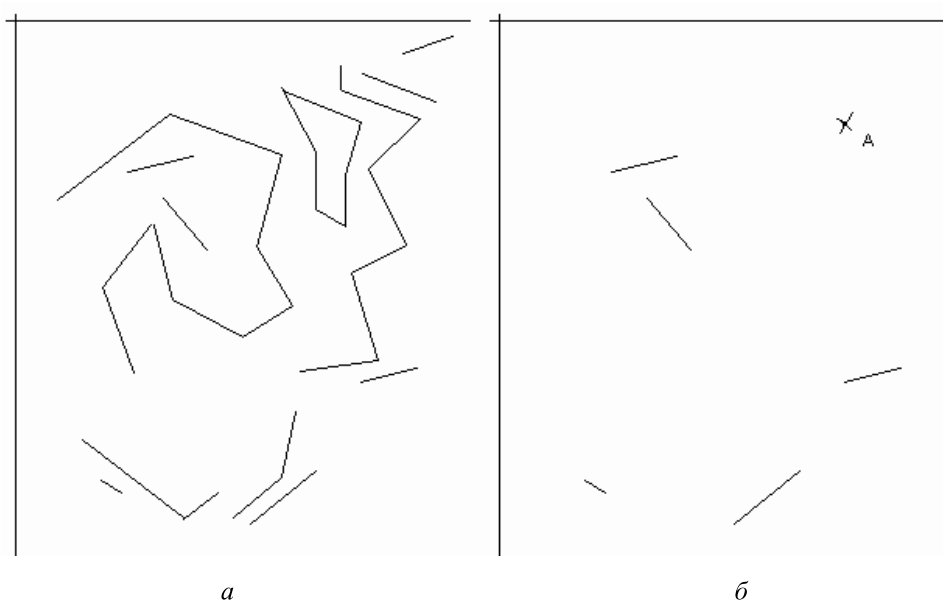


Рис. 6. На рис. а показано приклад розміщення відрізків, які потрібно нанести на заготовку за допомогою точок

На рис. б показано ситуацію, яка виникає (якщо послідовність руху знаходиться на основі мінімальної відстані) за декілька кроків до закінчення циклу проходження. Ми знаходимось в т. А. Як видно відрізки, які залишилися необробленими, розміщені на великій відстані (вона набагато більша, ніж їх власна довжина) від положення різального інструменту.

Переваги запропонованого тут алгоритму: серед наведених вище алгоритмів, траєкторія руху різального інструменту, яка створена на основі цього алгоритму, завжди буде відпрацьовуватись швидше (за однакової якості обробки).

Недолік запропонованого тут алгоритму: степінь оптимальності траєкторії залежить від конкретного зображення і його складності, тобто від коефіцієнта заповнення зображення (рис. 5).

Висновки

У статті було проаналізовано існуючі алгоритми формування траєкторії руху різального інструменту для точкового гравірування графічних зображень. Запропоновано модифікований алгоритм мінімальної відстані для створення траєкторії руху різального інструменту.

1. Фор А.. *Восприятие и распознавание образов.* – М.: Машиностроение. 1995. – С.51–55.
2. Горелик А.Л., Скрипка В.А. *Методы распознавания.* – М.: Высшая школа. 1997. – С.116–119.