УДК 004.932

О.Н. РОМАНЮК, О.В. МЕЛЬНИК, І.В. АБРАМЧУК
Вінницький національний технічний університет

ВИЗНАЧЕННЯ ТИПІВ КРОКОВИХ ПРИРОСТІВ ДЛЯ ПОБУДОВИ КОЛА НА ГЕКСАГОНАЛЬНОМУ РАСТРІ

Запропоновано типи крокових приростів для побудови кола на гексагональному растрі. Показано залежність вибору крокових приростів від октанту, в якому інтерполюється дуга кола.
Ключові слова: гексагональний растр, піксел, оцінювальна функція.

OLEKSANDR ROMANYK, OLEKSANDR MELNYK, IHOR ABRAMCHYK
Vinnytsia National Technical University

STEPPING TO DETERMINE THE TYPES INCREMENT FOR BUILDING CIRCLES ON THE HEXAGONAL RASTER

The problem of determining the types of stepping increments for constructing a circle when displayed on the screen, which includes a hexagonal pixels. Type Stepper increases proposed for constructing a circle on a hexagonal grid. The dependence of the choice of stepping increments sector which interpolates the arc of the circle. In discrete space is always some alternative choices dots that form the stepping trajectory. For maximum accuracy is selected from a set of points of the point, which is located closest to the ideal (continual) primitive. The paper defines types stepping increments for circle interpolation on a hexagonal grid.
Keywords: hexagonal raster, pixel, evaluation function.

Вступ

Для побудови графічних зображень використовують графічні примітиви. Найбільш поширеними є такі примітиви [1]: вектор (відрізок прямої), коло, еліпс. Оскільки графічні примітиви формуються в дискретному просторі (екранній системі координат), то траєкторія заміняється набором точок, які територіально розміщені ближче до ідеальної траєкторії. У дискретному просторі завжди є кілька альтернативних варіантів вибору точок, які формують крокову траєкторію. Для забезпечення максимальної точності вибирають із набору точок ту точку, яка найближче розміщена до ідеального (неперервного) примітиву.

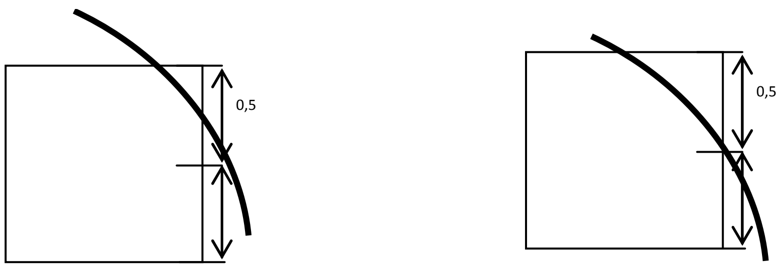
Ставиться задача визначення типів крокових приростів для побудови кола при відображенні на екрані, який включає гексагональні пікселі.

Аналіз досліджень та публікацій

Коло відноситься до графічних примітивів, які мають найбільшу з усіх питому вагу [1]. Найпоширенішим на сьогодні є використання для формування зображень квадратного растра, особливість якого полягає у розміщенні точок на екрані, які розташовані на однаковій відстані одна від одної. Зрозуміло, що при збільшенні роздільної здатності екрану якість формування кола зростає.

Для формування кіл найбільш поширеним є метод оцінювальної функції [3]. Згідно з методом траєкторія кола формується на основі деякої функції, знак якої визначає розміщення траєкторії до кола. Коли оцінювальна функція менше нуля, то крокова траєкторія знаходиться всередині кола, яке треба формувати. При розміщенні крокової траєкторії за колом оцінювальна функція більша нуля. Якщо ж оцінювальна функція дорівнює нулю, то точка траєкторії знаходиться на колі.

Всі алгоритми оцінювальної функції для побудови кола основані на зміні оцінювальної функції таким чином, щоб знак оцінювальної функції змінився на протилежний. Максимальна похибка при інтерполяції кола не перевищує пів кроку дискретизації (рис. 1).



а) б)

Рис. 1. Похибка 0,5 кроку при інтерполяції кола

У даному випадку при перетині колом дискретної решітки на відстані 0,5 дискрети вибирають ту, відстань до якої не перевищує вказану величину [3].

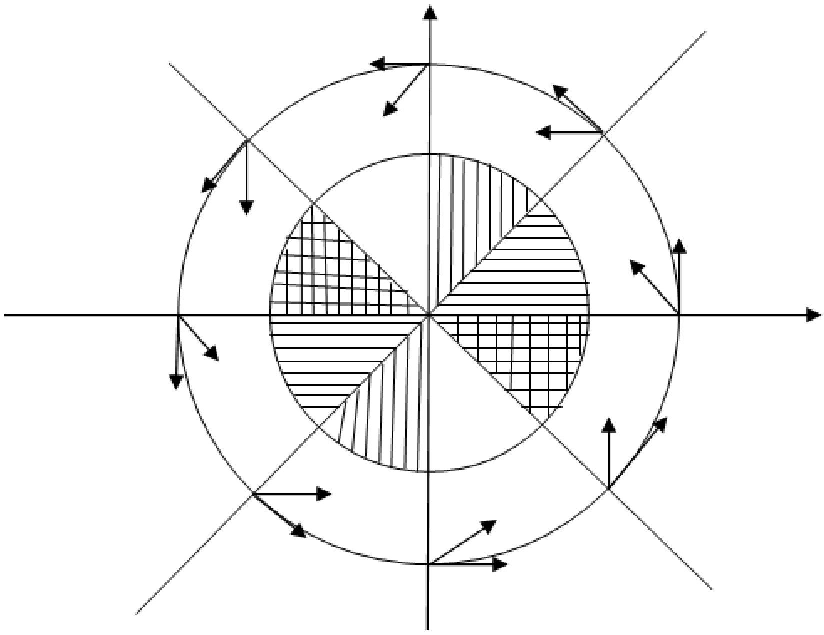


Рис. 2. Типи кроків для формування кола

Доведено, що для формування кола з максимальною точністю використовують горизонтальні, вертикальні, діагональні кроки[1, 3] (рис. 2). При використанні тільки горизонтальних і вертикальних кроків максимальна похибка може досягати відстані 1 дискрети.

На гексагональному растрі відстані на екрані між горизонтальними та вертикальними пікселями різні [2] (рис. 3).

Мета роботи – визначення типів крокових приростів при формуванні кола з максимальною точністю на гексагональному растрі.

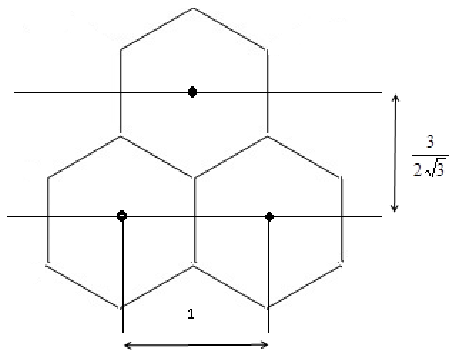


Рис. 3. Відстані між пікселями на гексагональному растрі

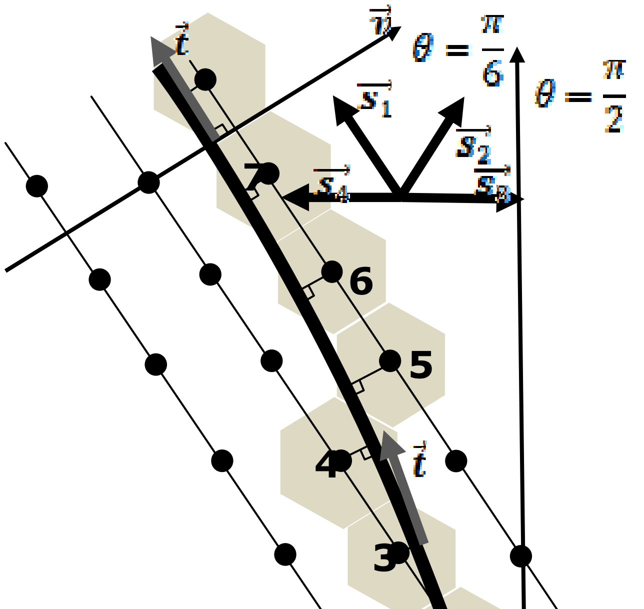


Рис. 4. Дуга кола на гексагональному растрі

Визначення типів крокових приростів при формуванні кіл на гексагональному растрі

Визначимо типи крокових приростів залежно від ділянок екрану (екранної системи координат).

Нехай потрібно відобразити на гексагональну сітку коло з центром $O(x_0, y_0)$ радіуса *R*. Розглянемо спочатку ділянку кола, що відповідає дузі $\theta \in \left[0, \frac{\pi}{6}\right]$, де θ – кут, утворений радіус-вектором кола з додатним напрямком осі *x* декартової системи координат. Нехай і надалі напрямок вектора визначає кут θ, утворений цим вектором з додатним напрямком осі *x*. Для вибраної ділянки кола вектор дотичної $\vec{t}$ до кола утворює кути $\theta \in \left[\frac{\pi}{2}, \frac{2\pi}{3}\right]$ (рис.4).

Вузлом називатимемо центр гексагонального пікселя. Приймемо, що відстань між сусідніми вузлами дорівнює 1. Відстанню від лінії (кола) до пікселя назвемо найкоротшу відстань від вузла до лінії (при умові, що лінія не має ширини).

Вважатимемо, що відстань є додатною, якщо вузол знаходиться праворуч від дотичної до лінії і від'ємною, якщо ліворуч. Для відображення на гексагональну сітку лінії (кола) зафарбовуються лише ті пікселі, для яких відстань до лінії найменша. Приймемо, що відображення лінії має товщину у один піксель. Якщо лінія неперервна, то пікселі зафарбовуються неперервно, тобто для кожного зафарбованого пікселя обов'язково має бути зафарбований лише один з шести сусідніх.

Доведемо, що якщо вибрано один з пікселів гексагональної сітки (початкова точка), то зафарбувати пікселі, що відображають задану дугу кола $\theta \in \left[0, \frac{\pi}{6}\right]$ можна рухаючись від початкової точки у напрямку годинникової стрілки, переходячи від пікселя до сусіднього пікселя з використанням **лише** переходів у напрямках $\overrightarrow{s_1}$ (що відповідає $\theta = \frac{2\pi}{3}$) і $\overrightarrow{s_2}$ (що відповідає $\theta = \frac{\pi}{3}$), причому перехід у напрямку $\overrightarrow{s_2}$ не може відбуватися двічі підряд.

Доведення. Дотична $\vec{t}$ до дуги кола $\theta \in \left[0, \frac{\pi}{6}\right]$ завжди утворює гострий кут з напрямком $\overrightarrow{s_1}$.

1. Якщо вузол знаходиться **з правого боку** від лінії (додатна відстань), то при переході до сусіднього вузла у напрямку вектора $\overrightarrow{s_1}$, відстань від вузла до лінії буде зменшуватись доти, доки пряма, що проходить через зазначені вузли не перетне дугу кола. Так відбувається для послідовностей вузлів 1, 2 і 5, 6, 7 (рис. 4). Рух у всіх інших напрямках $\overrightarrow{s_2}$, $\overrightarrow{s_3}$, $\overrightarrow{s_4}$ у цьому випадку буде лише збільшувати відстань від вузла до дуги кола, оскільки ці напрямки утворюють більші кути з напрямком $\vec{t}$.

Розглянемо (рис.4.) рух у напрямках $\overrightarrow{s_2}$, $\overrightarrow{s_3}$, з початкового вузла (0) віддаляє від лінії (враховано взаємне розміщення і можливі кути між векторами $\vec{t}$, $\overrightarrow{s_2}$, $\overrightarrow{s_3}$). Для руху у напрямку $\overrightarrow{s_4}$ можливі випадки: а) виконано перехід у вузол (0) з вузла (-2) (напрямок $\overrightarrow{s_1}$), тоді подальший рух у цьому напрямку до вузла (1) буде тільки зменшувати відстань до дуги; б) виконано перехід у вузол (0) з вузла (-1) (напрямок $\overrightarrow{s_2}$). Раніше проаналізовано, що відстань до дуги від вузла (0) менша за відстань від вузла (4).

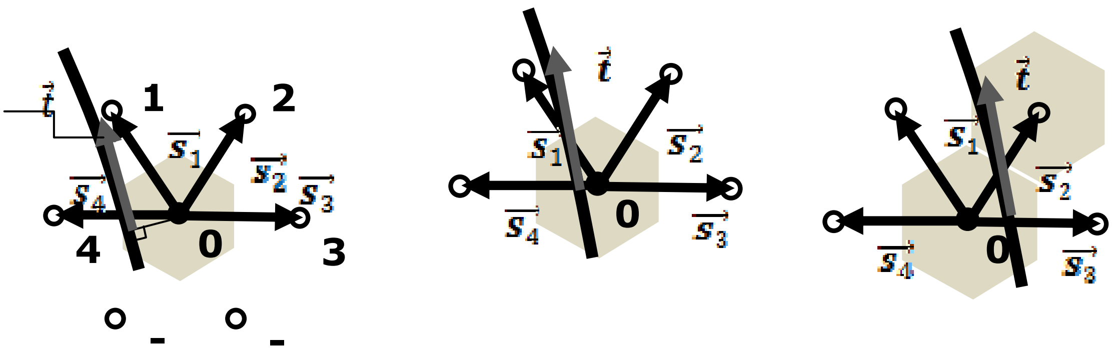


Рис. 5. Варіанти вибору кроків

У випадку, коли дуга кола перетинає відрізок, що з'єднує сусідні вузли у напрямку $\overrightarrow{s_1}$, то вузол, що знаходився зліва від дуги перейде (в напрямку $\overrightarrow{s_1}$) у вузол, що буде розташований справа від дуги, причому відстань до дуги буде завжди менше за 0.5. Те, що відстань мінімізується у напрямку $\overrightarrow{s_1}$ випливає з того, що вектори $\overrightarrow{s_2}$, $\overrightarrow{s_3}$, $\overrightarrow{s_4}$ утворюють з напрямком дотичної $\vec{t}$ більші кути (рис.5).

2. Якщо вузол знаходиться **зліва** від дуги, то можливі два випадки:

а) при переході у напрямку $\overrightarrow{S_1}$ новий вузол залишиться найближчим до дуги (послідовність вузлів 3, 4 на рис.5), переходи в напрямках $\overrightarrow{S_3}$, $\overrightarrow{S_4}$ все одно неможливі, так як вони утворюють більші кути з дотичною $\vec{t}$;

б) якщо при переході у напрямку $\overrightarrow{S_1}$ вузол **не** буде найближчим до дуги кола, то тоді перехід у напрямку $\overrightarrow{S_2}$ дасть нам найближчий вузол, так як відстань від такого вузла до кола буде завжди меншою за 0.5 (рис.4 та послідовність вузлів 4, 5 на рис.5). При цих умовах перехід $\overrightarrow{S_4}$ забезпечить більшу похибку, ніж при переході за $\overrightarrow{S_1}$, а перехід до сусіда праворуч (напрямок $\overrightarrow{S_3}$) дає відстань, що перевищує граничне значення 0.5. Так у найкращому випадку, значення близькі до граничного 0.5 при переході праворуч, можливі лише коли дотична $\vec{t}$ до дуги утворює кут $\theta = \frac{\pi}{2}$ і дуга проходить по дотичній через праву границю гексагону, але і тоді дуга не перетинає правий сусідній піксель, тому відстань від правого вузла більше за 0.5 і не є найменшою.

Зазначимо, що два переходи у напрямку $\overrightarrow{S_2}$, виконати підряд неможливо, так як перший з них виконується у випадку 2.б): дуга знаходиться справа від вузла і тому другий перехід збільшить відстань від дуги на відстань, що перевищує граничні 0.5 (враховано допустимі межі кута дотичної $\vec{t}$).

Розглянемо тепер дугу кола, що відповідає $\theta \in \left[-\frac{\pi}{6}, 0\right]$, вона симетрична дузі $\theta \in \left[0, \frac{\pi}{6}\right]$, тому у напрямку обходу кола проти годинникової стрілки оптимальні напрямки переходу $\overrightarrow{S_1}$ і $\overrightarrow{S_2}$ зберігаються.

Об'єднаємо отримані результати: для дуги кола $\theta \in \left[-\frac{\pi}{6}, \frac{\pi}{6}\right]$ напрямками побудови кола є виключно напрямки переходу $\overrightarrow{S_1}$, $\overrightarrow{S_2}$, причому два переходи у напрямку $\overrightarrow{S_2}$, виконати підряд неможливо.

Повернемо дугу кола $\theta \in \left[-\frac{\pi}{6}, \frac{\pi}{6}\right]$ на кут $\frac{\pi}{3}$ проти годинникової стрілки. В результаті отримаємо дугу $\theta \in \left[\frac{\pi}{6}, \frac{\pi}{2}\right]$, причому напрямки переходу $\overrightarrow{S_1}$, $\overrightarrow{S_2}$ перейдуть, відповідно, у напрямки, що відповідають кутам $\theta = \frac{2\pi}{3}, \theta = \pi$.

Використаємо тепер симетрію кола відносно напрямків, паралельних осям декартової системи координат. З попередніх досліджень випливає, що у кожному з секторів кола $\theta \in \left[-\frac{\pi}{6} + \frac{\pi}{3}n, \frac{\pi}{6} + \frac{\pi}{3}n\right]$, $n = 0..5$ існує лише по два напрямки переходу, причому по одному з них виконати два переходи підряд неможливо.

Висновки

У роботі визначено типи крокових приростів для інтерполяції кола на гексагональному растрі. Доведено залежність вибору типів крокових приростів від октанту в якому інтерполюється дуга кола.

Література

1. Пєтух А.М. Інтерполяція в задачах контурного формоутворення : [монографія] / Пєтух А.М., Обідник Д.Т., Романюк О.Н. – Вінниця : УНІВЕРСУМ-Вінниця, 2007. – 103 с.
2. Романюк О. Н. Особливості гексагональної моделі піксела / О. В. Мельник, О. Н. Романюк // Вимірювальна та обчислювальна техніка в технологічних процесах : міжнародний науково-технічний журнал. – Хмельницький : ХНУ, 2014. – № 1 (46). – С. 91–95.
3. Романюк О.Н. Використання методу оцінювальної функції для задач антиаліайзингу / О. Н. Романюк, М. С. Курінний // Сборник научных трудов НГУ. – Дніпропетровськ : НГУ, 2004. – № 19. – Том 2. – С. 200–208.
4. Wuthrich C. A. AnAlgorithmic Comparison Between Squareand Hexagonal-based Grid / C. A. Wuthrich, P. Stucki // CVGIP: Graphical Modelsand Image Processing. – 1999. – Vol. 53. – P. 324–339.

Отримана/Received : 3.5.2017 р. Надрукована/Printed :9.6.2017 р.
Рецензент: д. т. н., проф. Павлов С. В.

УДК: 004.932

Н.С. СВИРНЕВСКИЙ
Хмельницкий национальный университет

РАЗРАБОТКА АРХИТЕКТУРЫ ПРОГРАММЫ ДЛЯ РЕШЕНИЯ ЗАДАЧ, СВЯЗАННЫХ С АФФИННЫМИ ПРЕОБРАЗОВАНИЯМИ ПРОСТРАНСТВА

В статье описывается процесс создания архитектуры программы для решения геометрических задач на основе математического аппарата аффинных преобразований, отталкиваясь от элементарных графических возможностей, предоставляемых C++WinAPI приложением. Программа тестирована на примере решения задачи сканирования рупорной антенны. Продемонстрировано многообразие возможностей геометрических преобразований и эффективность управления ими.

Ключевые слова: программа, архитектура, аффинные преобразования, матрица, антенна.

N.S. SVIRNEVSKYI
Khmelnytskyi National University

DEVELOPMENT OF POGRAM ARCHITECTURE FOR SOLVING THE PROBLEMS ASSOCIATED WITH AFFINE TRANSFORMATION OF THE SPACE

This article describes creation of program architecture for solving of geometric problems based on the mathematical apparatus of affine transformation, starting from the basic graphics capabilities provided by C ++ WinAPI application. The program was tested on an example of solving the problem of scanning horn antenna. It demonstrated the variety of possible geometric transformations and their management efficiency.

Keywords: program, architecture, affine transformation, matrix, antenna.

Введение

Архитектура программы – это ее организация, воплощенная в компонентах (модулях), объединенных для выполнения определенной функции. Различают стандартные модули (библиотеки), входящие в язык программирования и пользовательские модули (программы), предназначенные для упрощения работы программистов.

Эта статья посвящена разработке программы, предназначенной для решения геометрических задач на основе аффинных преобразований пространства. К таким задачам, прежде всего, относятся – создание геометрических объектов, визуализация и обработка изображений, включая создание диалогов для поступления исходной информации и управления результатами.

Анализ исследований и публикаций

Известно большое количество стандартных модулей, обеспечивающих создание и обработку изображений [1, 2], решение многих задач в которых реализуется через аффинные преобразования [3]. При этом в них ограничены возможности для решения ряда нестандартных задач [4, 5], так как аффинные преобразования инкапсулированы в функциях и командах и предназначены лишь для конкретных действий по визуализации и обработке изображений.

На сегодняшний день, несмотря на многообразие стандартных графических библиотек и программных систем, по-прежнему, остается актуальной проблема создания эффективных пользовательских программных модулей для решения задач универсального характера, поскольку многие научные разработки выполняются именно с помощью их.

Формулирование цели

Разработать архитектуру программы для решения задач, связанных с аффинными преобразованиями пространства. Описать процесс создания программы на базе математического аппарата аффинных преобразований 2D и 3D пространства, отталкиваясь от элементарных графических