

Кроме вышеперечисленных классов базовый класс PCB содержит векторные массивы элементов, электрических цепей и посадочных мест.

Роль конвертора при преобразовании данных полученных из САПР KiCAD во внутреннее представление данных УИ САПР служат функция чтения данных «READ» и функция записи в файл «OUT».

Особое внимание следует уделить классу POS_PLACE, описывающего понятие посадочного места. В структуре данных KiCAD отсутствует разбиение печатной платы на посадочные места. Автоматическое размещение производится непрерывным методом по принципу группировки однотипных элементов на минимально близкое расстояние друг от друга. Однако, в ходе разработки ПО, более удобным и эффективным способом размещения элементов на плате был принят способ деления коммутационного поля (т.е. площади печатной платы) на четко определенные участки (посадочные места). Причем количество посадочных мест должно быть больше или равно количеству элементов на плате. За разбиение печатной платы на посадочные места и формирование списка этих мест с указанием координат центра и отвечают функции класса POS_PLACE.

Библиотека алгоритмов размещения включает в себя последовательный эвристический алгоритм; вероятностный алгоритм; генетический алгоритм; итерационный алгоритм парных перестановок. Алгоритмы размещения элементов реализованы в виде динамически подключаемых библиотек (DLL). Вопросы расширения и использования библиотеки программных модулей алгоритмов размещения частично рассмотрены в [4, 5].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. <http://ru.wikipedia.org/wiki/Kicad>.
2. http://www.lis.inpg.fr/realise_au_lis/kicad.
3. <http://www.pcad.ru/>.
4. Литвиненко В.А., Рябов О.В. Программная среда визуальной разработки программного обеспечения САПР // Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». – Таганрог: Изд-во ТРТУ, 2006, № 8(63). – С. 127-131.
5. Литвиненко В.А., Рябов О.В. Разработка учебно-исследовательской САПР печатных плат на основе САПР KiCAD // Труды Международных научно-технических конференций «Интеллектуальные системы» (AIS'06) и «Интеллектуальные САПР» (CAD-2006). Научное издание в 3-х томах. – М.: Физматлит, 2006, Т.2. – С. 103-110.

УДК 681.3.06

В.В. Лисяк, М.В. Лисяк

МЕТОД ПОИСКА РЕЗУЛЬТИРУЮЩЕЙ МАТРИЦЫ КОМПОЗИЦИИ ИНТЕРАКТИВНЫХ ГЕОМЕТРИЧЕСКИХ ПРЕОБРАЗОВАНИЙ В САПР*

Многие задачи, решаемые геометрическим процессором САПР, сводятся к следующей формулировке. Даны два объекта, причём известно, что один из объектов получен путём применения к другому объекту композиции аффинных преобразований. Требуется вычислить матрицу преобразования, которая трансформировала один объект в другой [1-4]. Ниже иллюстрируется постановка такой задачи:

* Работа выполнена при поддержке: РФФИ (грант № 08-01-00473), РНП 2.1.2.3193, РНП 2.1.2.2238, г/б № Т.12.8.08.

Известны матрицы: P_1 , P_2 и их взаимосвязь $P_1 * T = P_2$. Неизвестна матрица T . Решение этой задачи сводится к поиску обратной матрицы к матрице P_1 и умножения её на матрицу P_2 [1].

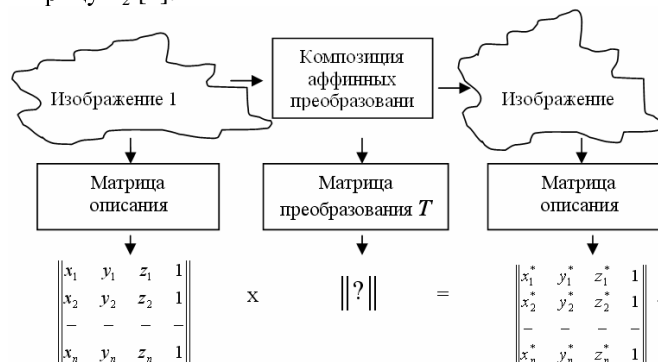


Рис. 1. Постановка задачи вычисления матрицы преобразования

Задачи, в сформулированной постановке, возникают при редактировании или преобразовании в диалоговом режиме объектов геометрического моделирования в САПР. При этом обе модели представляются в матричной форме в виде вектор-строк положения характерных точек, заданных в однородных координатах. Задача решается путём вычисления матрицы преобразований, трансформирующей исходный объект посредством цепочки аффинных преобразований. Однако известно, что обратную матрицу имеет только квадратная матрица. В связи с этим возникает вопрос, что делать, если исходная матрица не является квадратной.

Цели работы:

1. Разработать алгоритм решения задачи для матриц, содержащих произвольное число точек координат.
2. Разработать программу со средствами интерактивной и автоматизированной экспериментальной проверки предлагаемого алгоритма.
3. Предусмотреть экспорт найденной матрицы преобразования в программные продукты пользователей.

Исходные данные и требования к решению поставленной задачи:

1. Изображения представляются в трёхмерной области.
2. Размерность матрицы описания изображений $N \times 4$.
3. Для описания изображений и выполнения преобразований использовать однородные координаты.
4. Преобразования изображений должны выполняться в классе аффинных преобразований.

Теоретическое обоснование работы базируется на аппарате матричной алгебры и аппарате аффинной геометрии с использованием однородного координатного преобразования.

Аффинное преобразование – геометрическое преобразование плоскости или пространства, которое можно получить, комбинируя движения, зеркальные отображения и гомотетии в направлениях координатных осей. Гомотетия – преобразование подобия.

Структурная схема алгоритма, удовлетворяющая перечисленным целям и требованиям, показана на рис. 2.

Основной процедурой в решении указанной задачи является вычисление обратной матрицы. Показывается, что для вычисления матрицы преобразования

трёхмерного объекта, достаточно выделить в матрице одного объекта любую подматрицу 4×4 , найти к ней обратную подматрицу и умножить её на аналогичную по месту положения подматрицу 4×4 в матрице, описывающей второй объект.

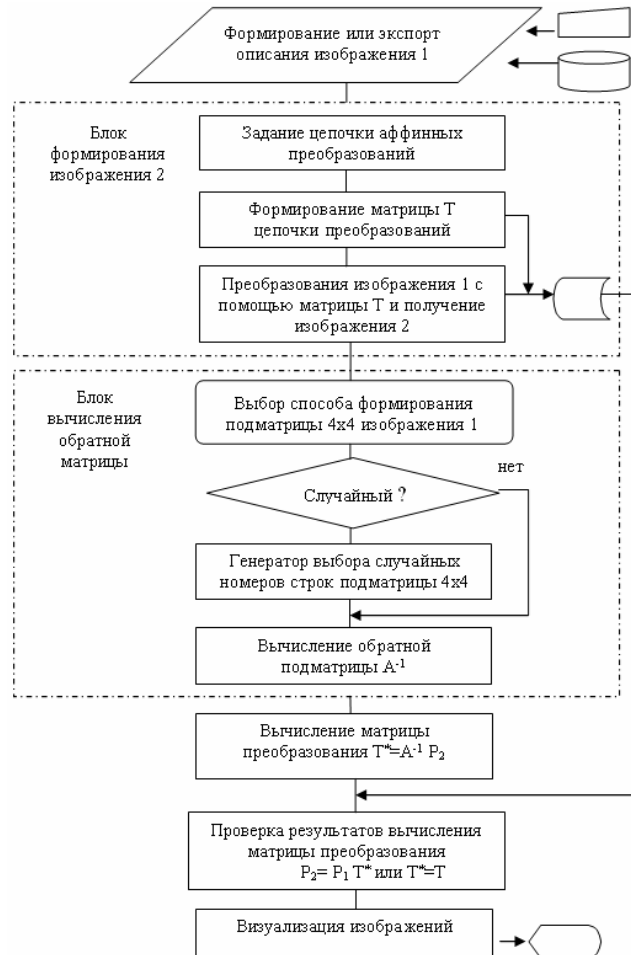


Рис. 2. Структурная схема алгоритма вычисления матрицы преобразования

Координаты точек в матрицах P_1 и P_2 представлены в однородных координатах. Представление двумерного вектора трёхмерным или в общем случае n -мерного вектора $(n+1)$ -мерным вектором называется однородным координатным преобразованием, которое выполняется в $(n+1)$ -мерном пространстве, а конечные результаты в n -мерном пространстве получаются с помощью обратного преобразования – операции нормализации, т.е. деления координат на значение однородной координаты [5]. При выполнении геометрических преобразований в двумерном пространстве операция деления не требуется, так как однородная координата равна 1.

Однородное координатное преобразование приводит к матрице преобразования размера 4×4 , что позволяет выполнить все базовые операции для трёхмерной области:

1. Все виды масштабирования.
2. Все виды отображений.
3. Произвольный перенос объекта.

4. Все виды сдвиговых операций.
5. Все виды вращений.

Преобразования в перспективе.

Блок формирования изображения 2 (см. рис. 2) предназначен для получения изображения в классе аффинных преобразований. Для этого задание цепочки преобразований выполняется в соответствии с аффинными, базовыми преобразованиями, обобщённая матрица которых имеет вид:

$$\left\| \begin{array}{ccc|c} a_x & c_1 & c_2 & p \\ c_4 & a_y & c_3 & q \\ c_5 & c_6 & a_z & r \\ \hline t_x & t_y & t_z & s \end{array} \right\| \rightarrow \left\| \begin{array}{cc|cc} 3 \times 3 & & 3 \times 1 & \\ \hline 1 \times 3 & & 1 \times 1 & \end{array} \right\| .$$

Матрица 3x3 осуществляет все виды масштабирования, сдвига, отображения и вращения. Матрица-строка 1x3 выполняет любые комбинации переносов изображения, матрица-столбец 3x1 все возможные фокусные преобразования в перспективе, а скалярный элемент s выполняет общее пропорциональное изменение масштаба. В соответствии с этим в таблице приведены аффинные базовые преобразования, используемые в процессе построения композиции преобразований.

На выходе блока формирования изображения гарантировано, что изображение 2 получено из изображения 1 в классе аффинных преобразований. Одновременно с заданием цепочки преобразований формируется результирующая матрица преобразования T , которая потребует в дальнейшем для сравнения с вычисленной матрицей преобразования T^* .

В блоке вычисления обратной матрицы выбор подматрицы 4x4 из матрицы P_1 изображения 1, может выполняться случайным образом, что полезно в режиме исследования программы, а также задаваться интерактивно или в режиме шаблона. Режим шаблона жестко задаёт подматрицу 4x4, например, первые четыре строки матрицы P_1 . После всестороннего исследования программы, режим случайного выбора подматрицы 4x4 можно исключить.

Проверка правильности вычисления матрицы преобразования T^* выполняется сравнением её с матрицей T , либо вычислением $P_2 = P_1 * T$ и сравнением P_2 с сформированной матрицей изображения 2.

Рассмотренный алгоритм вычисления матрицы преобразования справедлив, если преобразование одного объекта в другой выполняется в рамках аффинной геометрии, т.е. когда свойства объектов сохраняются при любых аффинных преобразованиях. Преобразование плоскости или пространства является аффинным, если его можно получить, комбинируя движения, зеркальные отображения и гомотетии (преобразования подобия) в направлении координатных осей. В связи с этим операции редактирования, связанные с добавлением или удалением точек объекта, недопустимы, так как приводят к совершенно другому объекту. Основные базовые операции преобразования (редактирования) изображений или геометрических моделей удовлетворяют аффинному преобразованию, поэтому рассмотренный алгоритм можно применять в подсистемах машинной графики и геометрического моделирования.

Программа реализации предлагаемого алгоритма написана в среде программирования Borland C++ Builder 6.0. и позволяет использовать следующие режимы работы:

1. Формирование исходного изображения для плоскости и пространства.
2. Интерактивное и случайное формирование композиции аффинных преобразований.

3. Интерактивный и случайный выбор подматриц исходной матрицы для вычисления обратной матрицы.
4. Выполнение проверки правильности вычисления обратной матрицы и визуализация изображений.

Таблица

Аффинные базовые преобразования

| Масштабирование | | |
|--|-----------|--|
| масштабирование по оси X | $e_x=$ | $M(E) = \begin{bmatrix} e_x & 0 & 0 & 0 \\ 0 & e_y & 0 & 0 \\ 0 & 0 & e_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| масштабирование по оси Y | $e_y=$ | |
| масштабирование по оси Z | $e_z=$ | |
| Переносы | | |
| перенос по оси X | $t_x=$ | $M(T) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$ |
| перенос по оси Y | $t_y=$ | |
| перенос по оси Z | $t_z=$ | |
| Перспективные преобразования | | |
| однофокусное с сходом на оси X | $p_x=$ | $M(Пер.) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| однофокусное с сходом на оси Y | $p_y=$ | |
| однофокусное с сходом на оси Z | $p_z=$ | |
| Отображения | | |
| относительно плоскости X=0 | $b_x=$ | $M(Om.) = \begin{bmatrix} -b_x & 0 & 0 & 0 \\ 0 & -b_y & 0 & 0 \\ 0 & 0 & -b_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ |
| относительно плоскости Y=0 | $b_y=$ | |
| относительно плоскости Z=0 | $b_z=$ | |
| относительно начала координат | | |
| Вращения | | |
| Вращение вокруг оси X | $\alpha=$ | $M(R(X, \alpha)) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \\ 0 & 0 & 0 \end{bmatrix}$ |
| Вращение вокруг оси Y | $\alpha=$ | $M(R(Y, \alpha)) = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ |
| Вращение вокруг оси Z | $\alpha=$ | $M(R(Z, \alpha)) = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \\ 0 & 0 & 0 \end{bmatrix}$ |
| Вращение вокруг оси, проходящей через начало координат | $\alpha=$ | есть |
| Вращение вокруг произвольной оси | $\alpha=$ | есть |

Отличительной особенностью программы является слабая зависимость времени работы программы от размерности исходного изображения, т.к. основная процедура нахождения матрицы преобразования выполняется для подматрицы 4x4 с последующим умножением на матрицу Nx4. Поэтому можно считать, что трудоёмкость программы пропорциональна времени перемножения двух матриц размерностью Nx4.

Программа может использоваться в геометрическом процессоре САПР для создания макросов типовых геометрических преобразований моделей проектируемых объектов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Никулин Е.А.* Компьютерная геометрия и алгоритмы машинной графики. – СПб.: БХВ-Петербург, 2003. – 560 с.
2. *Роджерс Д., Адамс Дж.* Математические основы машинной графики. – М.: Мир, 2001. – 604 с.
3. *Ласло Майкл.* Вычислительная геометрия и компьютерная графика на C⁺⁺: Пер. с англ. – М.: Бинум, 1997. – 304 с.
4. *Порев В.Н.* Компьютерная графика. – СПб.: БХВ-Петербург, 2002. – 432 с.
5. *Лисяк Н.К., Лисяк В.В.* Геометрическое моделирование в САПР: Учебное пособие. – Таганрог: Изд-во ТРТУ, 2005. – 82 с.

УДК 681.3

Н.Н. Орлов

ПОСТРОЕНИЕ СВЯЗЫВАЮЩИХ ДЕРЕВЬЕВ С РАЗНОЙ ШИРИНОЙ ФРАГМЕНТОВ СОЕДИНЕНИЙ*

Введение. Технология производства интегральных схем (ИС) продолжает своё стремительное развитие и в ближайшие годы размер минимального топологического элемента может достигнуть 30 нм., а количество транзисторов у сверхбольших ИС (СБИС) уже достигает 10^7 . Постоянно уменьшающиеся линейные размеры элементов ИС приводят к уменьшению мощности каждого активного компонента в отдельности и увеличению потребляемой мощности ИС в целом, за счет увеличения количества этих компонентов на кристалле. Это привело к тому, что задержки сигналов на межсоединениях доминируют над задержками на транзисторах, а рассеиваемая мощность становится определяющим фактором.

Традиционно, задача оптимизации длины соединений решается при синтезе их топологии на этапе глобальной трассировки путём построения кратчайших покрывающих деревьев или деревьев Штейнера. Если ширина всех соединений одинакова, то их суммарная длина и занимаемая ими площадь, как критерии качества, являются идентичными и отличаются только умножением на константу – ширину проводника. Однако, как известно, ёмкость плоского конденсатора определяется как

$$C = \epsilon \cdot \epsilon_0 \cdot S / d, \quad (1)$$

* Работа выполнена при поддержке: РФФИ (гранты № 07-01-00174, № 06-01-00272), РНП 2.1.2.3193, РНП 2.1.2.2238.