

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ЛИНЕЙНО-ВРАЩАТЕЛЬНОГО ПЕРЕМЕЩЕНИЯ СЛОЖНЫХ СИМВОЛОВ ПОДВИЖНЫХ ОБЪЕКТОВ НА БАЗЕ АФФИННЫХ ПРЕОБРАЗОВАНИЙ

*М.И. Васюхин, доктор технических наук
А.М. Касим, кандидат технических наук
В.В. Долынный, аспирант*
В.Ю. Трохименко, студент
e-mail: trokhymenkoviktor@gmail.com*

Рассмотрены вопросы генерации линейного перемещения сложного символа подвижного объекта на шаг, а также его поворота на определенный угол. Описаны существующие методы организации комбинированного линейно-вращательного перемещения сложного символа объекта, сделана оценка их недостатков. Для поворота растровых и векторных символов использован метод синусно-косинусных преобразований.

Ключевые слова: линейно-вращательное перемещение, сложный символ объекта, искажение образа, метод синусно-косинусных преобразований, аффинные преобразования

В настоящее время существуют системы, обеспечивающие в реальном режиме времени отображение на дисплее пространственного перемещения трактора или комбайна по полю и информирование фермера о необходимости увеличения или уменьшения расхода удобрений на том или ином участке поля.

Кроме применения в сельском хозяйстве, методы отображения линейно-вращательного перемещения подвижных объектов также используют и в аэронавигационных системах, в военных системах обороны, при составлении моделей, анимаций, создании мультипликационных приложений для определенного рода деятельности и т.д.

Комплекс методов организации линейно-вращательного перемещения сложного символа объекта рассматривался в работах ряда ученых [2, 5, 6, 8, 9].

Кратко остановимся на основных результатах теоретических и экспериментальных исследований, полученных в упомянутых источниках, касательно данной проблематики. Среди прочих программно-аппаратных методов и средств построения динамических сцен в работе [2] следует отметить метод базовых матриц и генератор вращения символов, реализованный в виде геометрического процессора. По сравнению с другими имеющимися методами, программно-аппаратный метод быстрой

* Научный руководитель – доктор технических наук, профессор М.И. Васюхин

© М.И. Васюхин, А.М. Касим,
В.В. Долынный, В.Ю. Трохименко, 2015

реализации плавного вращения обеспечил лучшее отображение эволюций сложных изображений символов, осуществляемых синхронно с изменением ориентации объектом, при размерах базовых матриц более 24×24 . Работа [9] содержит реализацию методов построения средств синтеза и визуализации изображений динамических объектов в навигационном пространстве для центров оперативного управления в режиме реального времени. В частности, предложены методы описания изображений, архитектура и алгоритмы функционирования системы синтеза изображений в операционно-ориентированных системах координат, оптимальных для представления и обработки компонентов движения объектов.

В работе [8] для осуществления операции вращения предлагается модифицированный алгоритм поворота сложного символа, реализующий метод базовых матриц, который предусматривает хранение символа в квадратной матрице памяти, составляющей точечный микрорастр и имеющей размер более 8×8 , причем в матрице прорисовывается контур этого символа, а все точки, лежащие вне контура имеют прозрачный цвет. Основная идея данного подхода заключается в использовании базы данных символов (БДС), которая хранит их азимутальные изображения, т.е. заранее создаются и сохраняются в памяти изображения для каждого символа объекта, повернутые на все углы, определенные пользователем. Для обеспечения поворота на заданный угол из БДС выбирается необходимое изображение символа и индицируется на экран.

В [6] дальнейшее развитие получил метод базовых азимутально-ориентированных изображений растрового символа, который позволяет получить необходимое количество азимутально-ориентированных изображений сложного с расширенной атрибутикой символа путем преобразования по предложенным формулам минимального числа его базовых изображений, повернутых в пределах $0^\circ - 45^\circ$.

В [5] решена задача быстрого преобразования данных, обеспечивающих на экране эволюции множества движущихся объектов и их адекватного отображения, представляемого в виде символов на цветном картографическом фоне. Такая задача, в соответствии с которой часть вычислительных операций, осуществляемых ранее программным путем, выполняется аппаратно с помощью измененного геометрического процессора, названного в [2] генератором вращения. Опыт применения таких генераторов [2] показал значительные затраты времени на перестройку блока базовых матриц в условиях быстрой смены типов изображений символов. Для ликвидации отмеченного недостатка был предложен комбинированный программно-аппаратный метод представления эволюций сложных пространственных перемещений объектов, главная особенность которого состоит в том, что в качестве элементов памяти блока базовых матриц взяты бистабильные симметричные переключатели с памятью, которые имеют свойство сохранять своё состояние памяти при отключении питающих напряжений.

Недостатками существующих методов поворота изображений могут быть:

- недостаточная точность поворота;
- затраты времени на выполнение алгоритма, что приводит к несвоевременному выполнению самого метода;
- нагрузка системы, которая выражается в скорости отображения объекта на экране видеотерминала.

Цель исследований – осуществление линейно-вращательного перемещения сложного символа объекта и объяснение причин возникновения искажений.

Материалы и методика исследований. Для визуализации линейно-вращательного движения имеем сложный символ (рис. 1), который может быть представлен как в векторном, так и растровом виде, карту участка местности, среду программирования – Delphi.

При построении символов объектов, содержащих статические и динамические элементы, часто приходится иметь дело с ситуациями, когда они включают в себя целый ряд компонентов, отличающихся друг от друга только местоположением, ориентацией, размером, т.е. отдельные символы объектов обладают значительным геометрическим сходством.

В этом случае целесообразно описать один графический объект в качестве базового, а затем получать остальные требуемые объекты путем использования операций преобразования.

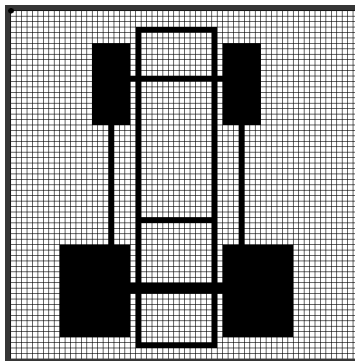


Рис. 1. Изображение матрицы точек, содержащей сложный символ объекта

С помощью операций преобразования можно выполнять следующие действия:

- перемещать объекты из одного места экрана в другое;
- создавать объект из более мелких элементов (составных частей);
- добавлять к существующему объекту новые элементы;
- увеличивать размер объекта для улучшения его наглядности или отображения более мелких деталей;
- уменьшать размер объекта для внесения, например поясняющих надписей или отображения на экране новых объектов;
- создавать движущиеся объекты.

Все представления эволюций символов можно выполнить с помощью трех базовых операций:

- переноса (перемещения) символа объекта;
- масштабирования (увеличения или уменьшения размеров) символа объекта;
- поворота символа объекта (употребляют также термины вращение, изменение ориентации).

Эти операции называются аффинными преобразованиями, при этом различают двумерные и трехмерные аффинные преобразования.

Поскольку для нашего случая область исследований ограничена 2D пространством, то рассмотрим лишь основные правила двумерных аффинных преобразований:

- прямые линии после преобразований остаются прямыми;
- параллельные прямые – параллельными;
- отношения деления отрезков остаются неизменными.

Процесс перемещения объекта, отображаемого на экране видеотерминала, складывается из двух элементарных составляющих: линейного перемещения и поворота. Более сложным является процесс вращения объекта (поворот на произвольный угол). Именно данный процесс требует значительных вычислительных ресурсов [2, 5, 6, 8].

Экспериментально установлено, что реакция на изменение направления поворота символа человеком-оператором находится в пределах $\pm 8^\circ - 12^\circ$ [3].

Задача состоит в том, чтобы с применением аффинных преобразований линейно переместить объект на определенный шаг, а также вращать его вокруг своей оси в поворотных точках траектории. Такая сцена представляется на экране видеотерминала, поэтому следует учитывать особенности экранной системы координат.

Результаты исследований. При переносе объекта необходимо знать координаты точки экрана, к которой привязывается выводимое изображение объекта. Один из способов переноса – это за каждым символом закрепить так называемую «ключевую точку», при этом в качестве абсолютной системы координат выбирается координатная сетка, пересечения строк и столбцов которой совпадают с пикселями экрана, а начало отсчета совпадает с его левым верхним углом. В случае представления символа объекта в растровом виде, его отображение осуществляется при помощи микрорастра (рис. 1). При этом предлагается относительная система координат, представляемая координатной сеткой, единичные клетки которой совпадают с элементами матрицы памяти символа. В ней в качестве «ключевой точки» выбирается верхний левый угол микрорастра [8].

Изображение на рис.1 сконструировано с помощью встроенного в среду Delphi редактора Image Editor.

Для визуализации линейного перемещения символа объекта, например на 10 шагов вверх, при использовании среды разработки Delphi, код операции будет представлен следующим образом:

$Image1.Top := Image1.Top - 10$,
где $Image1.Top$ – изменяемая позиция верхнего края изображения.

Для перемещения символа объекта вниз – то же самое, только со знаком «+».

Для перемещения влево, код операции будет выглядеть так:

$Image1.Left := Image1.Left - 10$,
где $Image1.Left$ – изменяемая позиция левого края изображения.

Для перемещения вправо – то же самое выражение, только с «+».

Зачастую при вращении графических объектов на определенный угол (A) (рис. 2) используется метод синусно-косинусных преобразований.

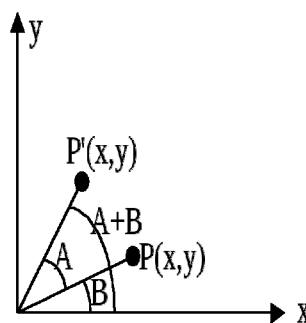


Рис. 2. Двумерный поворот

Математическое представление данного метода записывается в виде:

$$\begin{aligned}x &= r \cdot \cos B; \\y &= r \cdot \sin B,\end{aligned}$$

а новые координаты объекта вычисляются по формуле:

$$\begin{aligned}x' &= x \cdot \cos A - y \cdot \sin A; \\y' &= x \cdot \sin A + y \cdot \cos A.\end{aligned}$$

В матричной форме поворот записывается следующим образом:

$$M = \begin{bmatrix} \cos A & \sin A \\ -\sin A & \cos A \end{bmatrix}.$$

При этом движение считается положительным ($A > 0$), если вращение осуществлено против часовой стрелки; отрицательным – по часовой.

На рис. 3 приведена блок-схема предложенного алгоритма вращения символа объекта, с помощью которого можно осуществить его поворот на $45^\circ (\frac{\pi}{4})$ по часовой стрелке вокруг своей оси. В данном

алгоритме используются синусно-косинусные преобразования [4]. Согласно алгоритму каждый новый символ объекта поворачивается относительно своего предыдущего положения; после чего происходит выполнение команды для вывода на экран повернутого изображения символа. При данном алгоритме при каждом новом повороте следует искажение образа символа объекта.

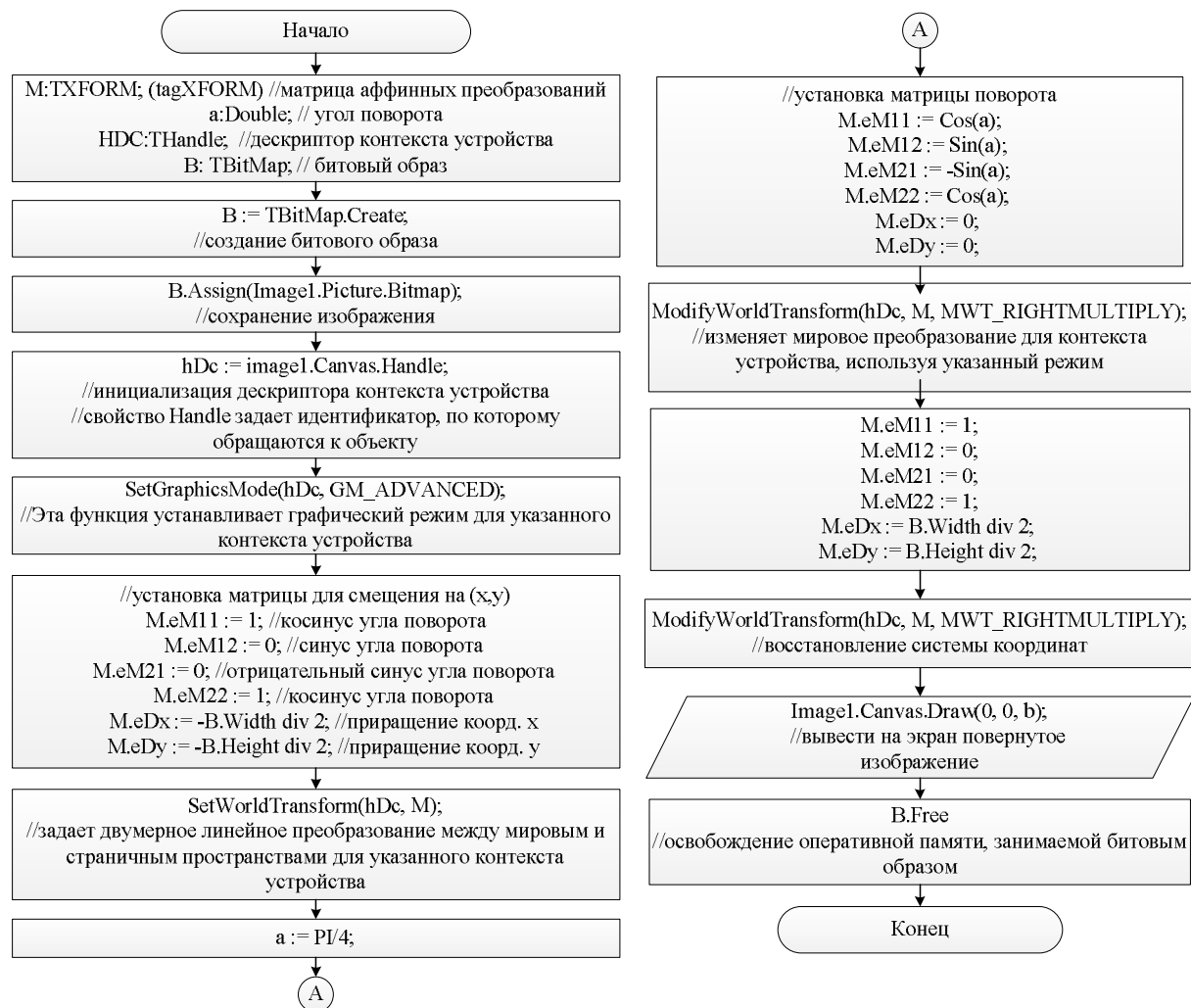


Рис. 3. Алгоритм вращения символа объекта

Если во время поворота последующее положение символа вычислять на основе координат его точек в предыдущем положении, то через некоторое время форма символа начнет искажаться. Это происходит в результате накопления ошибок, которые возникают в процессе преобразования координат. Чтобы этого не происходило, необходимо зафиксировать исходное положение символа, и координаты узловых точек его контура для каждого нового угла вращения вычислять по этому исходному положению. Кроме того, необходимо преобразовывать рассчитанные координаты в графические с учетом дискретности растровой сетки монитора [6].

Искажение формы символов также может быть связано с особенностями дисплея, который по своей структуре имеет пиксели не в

виде точки, а такие, которые по форме приближаются к прямоугольникам. В этом случае приходится дополнительно вводить коэффициенты, которые сокращают неравномерность между размерами пикселей по вертикали и горизонтали в соотношении 4:3 [6].

Итак, можно выделить две основные причины искажения изображения при повороте:

- меняется размер фигуры, куда это изображение вписано;
- меняются координаты точек.

В отличие от цифрового видео, где пиксели прямоугольные, в компьютерной графике пиксели квадратные. Это утверждение эквивалентно высказыванию, что видео отображается по-разному в горизонтальных и вертикальных направлениях.

У квадратного пиксела пропорции (Pixel Aspect Ratio) равны отношению 1:1, а у прямоугольного – соотношение сторон равно отношению ширины к высоте (W:H), причем W и H не равны [10].

Ширина (Width) и высота (Height) видеокadra измеряются в пикселах, и связаны друг с другом следующим соотношением:

$$\frac{\text{ширина (Width)}}{\text{высота (Height)}} = \frac{\text{DAR (пропорция устройства отображения)}}{\text{SAR (типовая пропорция пиксела)}}$$

где DAR – Display Aspect Ratio – описывает форму физически отображаемого на экране изображения,

SAR – Sample Aspect Ratio.

Например, в соответствии с промышленным стандартом «Industry Standard square pixels» для представления цифрового видео предполагается частота дискретизации, при которой пиксели – не квадратные. Они имеют пропорцию 768:767, которая является очень близкой к 1:1. Чтобы получить истинные пропорции пиксела при пересчете форматов, табличные значения нужно умножить на 768:767. С той же целью рекомендуется производить аналогичные вычисления в приложениях компьютерной графики.

Ниже представлено более детальное описание команд WinAPI, использованных при написании программы на языке программирования Object Pascal [11] и встречающихся в алгоритме на рис. 3.

function SetGraphicsMode (hdc: HDC, iMode: Integer):Integer – устанавливает графический режим для указанного контекста устройства. В ней hdc – дескриптор контекста устройства, iMode – определяет графический режим. Последний параметр может принимать одно из нижеследующих значений:

GM_COMPATIBLE – устанавливает графический режим, который совместим с 16-разрядным Windows;

GM_ADVANCED – устанавливает расширенный графический режим, который дает возможность преобразования мирового пространства. Это значение должно быть указано, если приложение будет установ-

ливать или модифицировать преобразование мирового пространства для указанного контекста устройства.

function SetWorldTransform (hdc: HDC; const p2: tagXFORM): LongBool – устанавливает двухмерное линейное преобразование между мировым пространством и пространством страницы для указанного контекста устройства. Это преобразование используется для масштабирования, поворота, сдвига или переноса выходных графических данных, где:

hdc – дескриптор контекста устройства;

p2 – указывает на структуру XFORM, которая содержит данные преобразования.

Если функция завершается успешно, возвращаемое значение отлично от нуля. Если же функция завершается ошибкой, возвращаемое значение равно нулю.

function ModifyWorldTransform (hdc: HDC; const p2: tagXFORM; p3: Cardinal): LongBool – изменяет преобразование мирового пространства для контекста устройства, используя указанный режим, где:

hdc – дескриптор контекста устройства;

p2 – указатель на структуру XFORM, используемую для модификации преобразования мирового пространства для данного контекста устройства;

p3 – определяет, каким образом данные преобразования модифицируют текущее преобразование мирового пространства. Этот параметр должен иметь одно из нижеследующих значений:

MWT_IDENTITY – сбрасывает текущее преобразование мирового пространства при помощи использования единичной матрицы. Если задается этот режим, структура XFORM, указанная при помощи p2 – игнорируется;

MWT_LEFTMULTIPLY – умножает текущее преобразование на данные в структуре XFORM. При этом данные в структуре XFORM становятся левым сомножителем, а данные для текущего преобразования – правым сомножителем;

MWT_RIGHTMULTIPLY – умножает текущее преобразование на данные в структуре XFORM. При этом данные в структуре XFORM становятся правым сомножителем, а данные для текущего преобразования – левым сомножителем.

В случае подачи символов в растровом виде, в зависимости от степени сложности, они задаются матрицами точек различной размерности, от 8×8 до 64×64, а в случае их представления в векторном формате – строятся из графических примитивов (точек, линий и дуг) по алгоритмам симметричного и несимметричного цифрового дифференциального анализатора, Бу и Брезенхема. Суть первого из этих алгоритмов является растеризация отрезка прямой между начальной и конечной точками путем интерполяции значений из интервалов $[(x_{start}, y_{start}), (x_{end}, y_{end})]$, исчисляемой для каждого x_i с помощью выражений: $x_i = x_{i-1} + 1/m$, $y_i = y_{i-1} + m$, где $\Delta x = x_{end} - x_{start}$, $\Delta y = y_{end} - y_{start}$ и $m = \Delta y / \Delta x$. Алгоритм Брезенхема определяет, какие точки двумерного

растра нужно закрасить, чтобы получить аппроксимацию прямой линии между двумя заданными точками. Алгоритм Ву – это модифицированный алгоритм Брезенхема, обеспечивающий разложение отрезка в растр со сглаживанием.

Для формирования динамических сценариев предусматриваются следующие операции: формирование и вывод визуальной статической информации, то есть картографического фона, и динамической, которая представляет перемещения подвижных объектов на нем.

Исследования [2, 5, 6] показали, что для отображения перемещений относительно простых символов в матрице 8x8 точек и при их небольшом количестве (3–5), вполне допустимо применять синусно-косинусные преобразования.

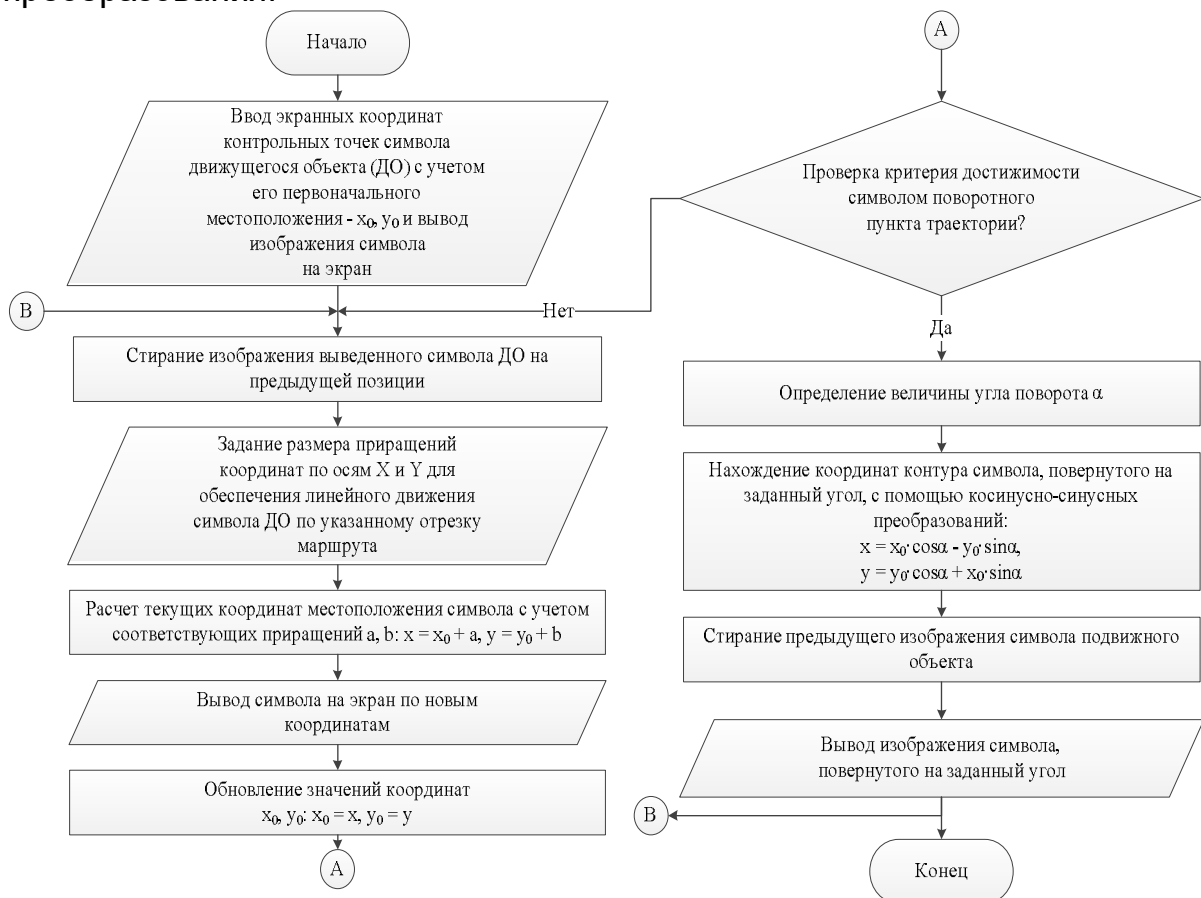


Рис 4. Алгоритм линейного перемещения и поворота сложного символа, реализующий метод синусно-косинусного преобразования

Для отображения линейного и вращательного перемещения сложного символа движущегося объекта с помощью метода синусно-косинусных преобразований предлагается алгоритм, который представлен на рис. 4.

Выводы

Проанализированы модели, методы, алгоритмы и средства визуализации линейно-вращательного перемещения сложного символа, движущегося в пространстве объекта. Предложены алгоритмы линейно-вращательного перемещения сложного символа объекта, представляемого как в растровом, так и в векторном форматах. Для линейного

перемещения и поворота на определенный угол исследуются двумерные аффинные преобразования. Выяснены причины возникновения искажений сложного символа объекта при реализации процедуры вращения и предложены приемы их устранения.

Список литературы

1. Аніскевич Л.В. Навігація і управління рухом безпілотних польових машин/Аніскевич Л.В., Войтюк Д.Г., Захарін Ф.М. – Ніжин: Видавець: ПП Лисенко М.М., 2012. – 96 с.
2. Васюхин М.И. Основы интерактивных навигационно-управляющих геоинформационных систем: Монография / М.И. Васюхин. – К.: Лири-К, 2006. – 536 с.
3. Венда В.Ф. Видеотерминалы в информационном взаимодействии (инженерно-психологические аспекты) / В.Ф. Венда. – М.: Энергия, 1980. – 198 с.
4. Вращение (поворот на заданный угол) векторного изображения [Электронный ресурс]. – Режим доступа: <http://www.programmersforum.ru/showthread.php?t=39977>
5. Капштык О.И. Методы и технологические средства представления и анализа воздушной обстановки в геоинформационных системах оперативного управления: дис. ... канд. техн. наук: 05.13.06 / Капштык Олег Иванович. – К., 2010. – 188 с.
6. Касим А.М. Методы и средства формирования динамических сценариев в навигационно-управляющих комплексах: дис. ... канд. техн. наук: 05.13.06 / Касим Аниса Мохаммадовна. – К., 2012. – 195 с.
7. Кравчук В.І. Теоретичні основи адаптації сільськогосподарських машин: монографія / В.І. Кравчук. – К: НАУ, 2005. – 208с.
8. Креденцар С.В. Методы и средства построения зрительных образов динамической обстановки в аэронавигационных геоинформационных системах реального времени: дис. ... канд. техн. наук: 05.13.06 / Креденцар Светлана Максимовна. – К., 2010. – 174 с.
9. Смолій В.В. Методи і засоби синтезу і візуалізації зображень динамічних об'єктів у реальному часі: монографія / В.В. Смолій. – Луганськ: Вид-во Східноукраїнського наці. уні-ту ім. Володимира Даля, 2005. – 176 с.
10. Соотношение сторон (Aspect Ratio) и разрешение кадра [Электронный ресурс]. – Режим доступа: <http://www.provegas.ru/forum/showthread.php?t=773>
11. WinAPI Reference [Электронный ресурс]. – Режим доступа: <http://winapi.freetchsecrets.com>

ПРОГРАМНА РЕАЛІЗАЦІЯ ЛІНІЙНО-ОБЕРТАЛЬНОГО ПЕРЕМІЩЕННЯ СКЛАДНИХ СИМВОЛІВ РУХЛИВИХ ОБ'ЄКТІВ НА БАЗІ АФІННИХ ПЕРЕТВОРЕНЬ

М.І. Васюхін, А.М. Касім, В.В. Долинний, В.Ю. Трохименко

Розглянуто питання генерації лінійного переміщення складного символу рухомого об'єкта на крок, а також його повороту на певний кут. Описано існуючі методи організації комбінованого лінійно-обертального переміщення складного символу об'єкта, зроблено оцінку

їх недоліків. Для повороту растрових і векторних символів використано метод синусно-косинусних перетворень.

Ключові слова: лінійно-обертальне переміщення, складний символ об'єкта, спотворення образу, метод синусно-косинусних перетворень, афінні перетворення

SOFTWARE IMPLEMENTATION LINE-ROTATIONAL MOVEMENT OF COMPLEX CHARACTERS MOVING OBJECTS BASED AFFINE TRANSFORMATIONS

M. Vasyuhin, A. Kasim, V. Dolyunny, V. Trokhymenko

The problems of generating linear movement of the movable object of a complex character on the pitch, as well as its rotation by a specific angle. Describes current methods of organizing a combined linear and rotational movement of the complex character of the object, made an assessment of their shortcomings. To rotate raster and vector characters used method of sine-cosine transforms.

Keywords: line-rotational movement, complex character of the object, the distortion of the image, the method of sine-cosine transformations, affine transformations

УДК 621.314

АНАЛІЗ ВПЛИВУ СТРУКТУРИ ПОВЕРХНІ ЗЕРНИНИ НА РОЗПОДІЛ ЕЛЕКТРИЧНОГО ПОЛЯ

К.В. Дубовенко, доктор технічних наук

Д.О. Захаров, асистент

Миколаївський національний аграрний університет

e-mail: zakharov.dim@gmail.com

Для визначення умов генерації озону коронним розрядом в об'ємі зернової маси проаналізовано структуру поверхні зернівок злакових культур. Обґрунтовано, що наявність неоднорідностей їх поверхні впливає на характеристики розподілу електричного поля в міжзернинних проміжках насипного збіжжя. Показано, що велика кількість неоднорідностей на поверхні зернівки пшениці призводить до значного підсилення напруженості електричного поля та об'єму області його збурення.

Ключові слова: структура поверхні, насипне зерно, напруженість електричного поля

Нині актуальним є питання підвищення ефективності та зменшення собівартості знезараження зернової продукції. Одним із способів