

УДК 681.327.22:003.6

И. М. Арсенин, Б. Б. Морозов, В. Г. Садов, И. Г. Таранцев  
(Новосибирск)

**МЕТОД ПОСТРОЕНИЯ ИЗОБРАЖЕНИЙ  
В СИСТЕМАХ ГЕНЕРАЦИИ ТИТРОВ**

Предложен ряд алгоритмов построения изображений в системах подготовки титров на базе персонального компьютера. Применение данных алгоритмов позволяет обеспечить высокую скорость генерации титров даже при одновременном использовании нескольких сложных трехмерных эффектов.

**Введение.** При подготовке телепрограмм, а также непосредственно в процессе вещания необходимо использовать титры, текстовые сообщения, символы и знаки. Для создания соответствующих сигналов и наложения их на видеоряд служат генераторы титров или знакогенераторы [1]. Основными критериями оценки знакогенераторов являются: количество шрифтов, одновременно доступных пользователю; разнообразие применяемых эффектов\*; скорость и удобство подготовки титров; возможность работы в прямом эфире.

Резкое удешевление персональных компьютеров с одновременным увеличением производительности и высокой степенью универсальности привело к буму вокруг создания на их основе систем видеопроизводства. Эксперименты показали, что производительности процессоров класса "Pentium-75" недостаточно для генерации сложных титров в реальном времени. Поэтому процесс генерации титров на базе персонального компьютера обычно разделяется на две части: подготовка титров и их анимация. Подготовка титров ведется в интерактивном режиме без жестких временных ограничений. По заданному тексту, выбранному шрифту и набору эффектов система подготовки титров строит изображение и сохраняет его в формате, удобном для дальнейшего воспроизведения в режиме реального времени. Система анимации титров отображает в реальном времени подготовленное изображение в соответствии с некоторым сценарием перемещения титров по экрану. Как правило, анимация титров осуществляется с помощью специализированных аппаратных средств, в то время как подготовка титров может вестись на стандартном оборудовании.

Целью данной работы является разработка алгоритма построения символа в системе подготовки титров, обеспечивающего время построения изображения не более 0,1 с. Такое время реакции системы обеспечивает интерактивный режим работы, когда за доли секунды все изображение изменяется одновременно с действиями пользователя.

**Построение изображения символа.** В большинстве существующих знакогенераторов используется фиксированный набор эффектов и фиксированный порядок их исполнения. Например, всегда вначале отображается тень, затем контур символа и, наконец, сам символ. Пользователь может изменить параметры эффектов, но не может повторить один и тот же эффект дважды (у символа только одна тень) или изменить порядок следования эффектов (тень

\* Эффектом в системах титрования называют метод построения цветного изображения по описанию символа.

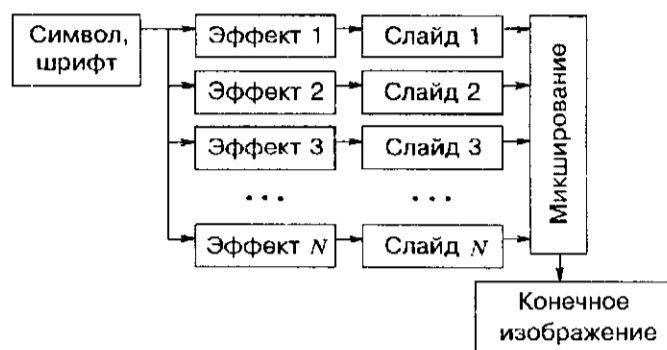


Рис. 1

от одного символа никогда не ложится на другой символ). Этот способ удобен для аппаратной реализации, но не обеспечивает достаточного разнообразия конечных изображений. Более современный подход подразумевает наличие некоторого набора базовых эффектов и многократное использование этих эффектов с разными параметрами в произвольном порядке (например, у символа может быть несколько теней с разной степенью прозрачности). Данный подход позволяет получить практически бесконечное множество конечных изображений даже при небольшом наборе базовых эффектов.

Процесс построения конечного изображения символа выглядит следующим образом. Пользователь задает произвольную последовательность эффектов (рис. 1). Каждый эффект является одним из базовых эффектов со своим набором параметров. В соответствии с эффектом строится слайд — промежуточное изображение с информацией о цвете и прозрачности в каждом пикселе. Конечное изображение является видимым «на просвет» образом всех слайдов.

Чтобы понять назначение основных алгоритмов предлагаемой статьи, необходимо рассмотреть некоторые характерные эффекты, применяемые при создании титров.

*Заливка цветом* подразумевает разделение всего изображения на две части (рис. 2). Одна часть (текст) закрашивается одним цветом, другая (фон) — другим (в роли фона обычно выступает внешний видеосигнал). Для этого эффекта необходимо создание битовой карты символа, задающей принадлежность пикселей либо тексту, либо фону. Наличие фильтрации требует задания в каждом пикселе байтового соотношения текст/фон для возможности осуществления плавного перехода текст/фон на границе символа.

*Обводка* — выделение контура символа отдельным цветом (рис. 3). Для этого эффекта необходима информация о минимальном расстоянии от точки до контура для всех точек внутри символа.



Рис. 2

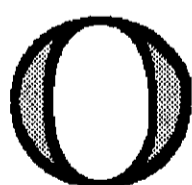


Рис. 3

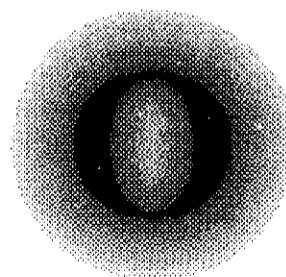


Рис. 4

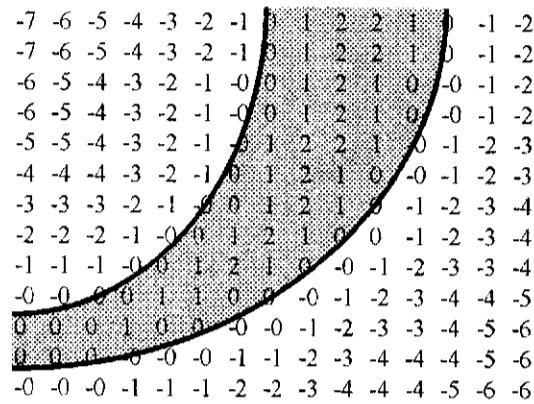


Рис. 5

**Сияние** — закраска пространства вокруг символа некоторым цветом, прозрачность которого пропорциональна расстоянию от точки до контура (рис. 4). Для этого эффекта необходима информация о минимальном расстоянии от точки до контура для всех точек вне символа.

**Реализация эффектов.** После подробного рассмотрения большого количества различных эффектов нами было отмечено, что для реализации большинства из них необходимо и достаточно создание маски минимального расстояния до края символа для всех точек внутри символа и вокруг него (рис. 5). Следовательно, можно применить схему с предварительным вычислением маски расстояний (рис. 6), что существенно сокращает время построения промежуточных изображений.

Кроме того, маску расстояний можно использовать для антиэлайсинга [2, 3]: интенсивность цвета пиксела на границе символа устанавливается в соответствии с отношением между площадью пиксела, накрываемой объектом, и площадью всего пиксела. Для модели «круглого» пиксела (рис. 7) возможно использование следующей функции фильтрации:

$$S(d) = 1 - \frac{\arccos(d) + \sqrt{1 - d^2}}{\pi},$$

где  $d$  — удаленность центра пиксела от края символа.

Для реализации эффекта заливки цветом, например, возможно использование следующего алгоритма ( $mask(x, y)$  — двумерный массив минимальных

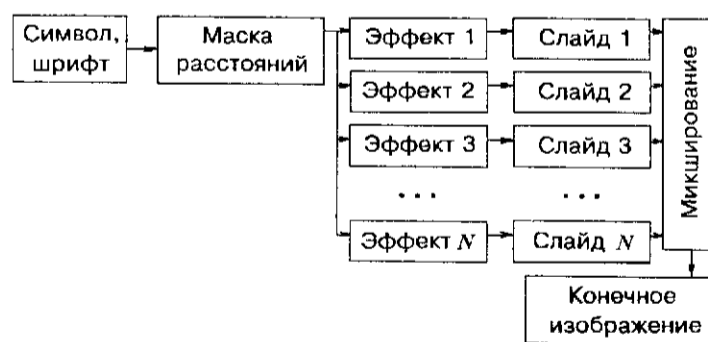


Рис. 6

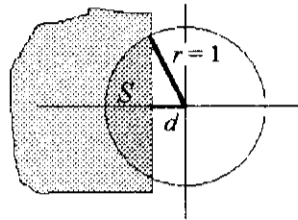


Рис. 7

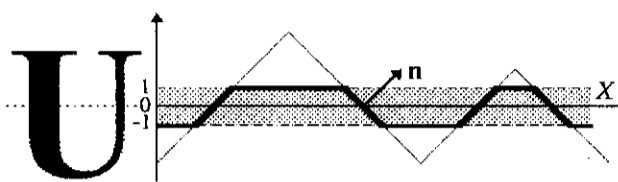


Рис. 8

расстояний от пикселей с координатами  $x, y$  до контура символа, процедура PutColor( $x, y, color$ ) записывает цвет в битовую карту символа):

```

{Алгоритм заливки цветом}
if mask (x, y) < 0.0 then
  PutColor (x, y, TextColor) {пиксел имеет цвет текста}
else if mask (x, y) > 1.0 then
  PutColor (x, y, FonColor) {пиксел имеет цвет фона}
else
  begin
    {цвет пиксела интерполируется цветами текста
    и фона пропорционально S(mask(x, y))}
    proportion = S (mask (x, y));
    color = FonColor * proportion +
           + TextColor * (1.0 - proportion);
    PutColor (x, y, color);
  end

```

**Реализация трехмерных эффектов.** Расстояние до края символа можно рассматривать как высоту: чем больше расстояние от точки до контура внутри символа, тем выше находится данная точка над уровнем моря. Соответственно чем больше расстояние до точки снаружи символа, тем ниже точка (рис. 8). По матрице высот можно построить нормаль к поверхности:

$$n = \frac{[v1 \times v2]}{[v1 \times v2]}, \quad v1 = (1, 0, dz1), \quad v2 = (0, 1, dz2),$$

где  $dz1$  и  $dz2$  — разница высот по горизонтали и вертикали соответственно (рис. 9).

В сферической системе координат нормаль к поверхности определяется особенно легко: отклонение от вертикали  $\theta$  практически всегда равно  $45^\circ$ , а угол  $\varphi$  в плоскости экрана (маски) легко вычисляется из формулы  $\operatorname{tg}(\varphi) = \frac{dz2}{dz1}$ .

Зная вектор нормали, можно вычислить цвет данной точки (рис. 10):

$$c = \left( c_a + c_d \frac{1 + \cos(\alpha)}{2} \right) (1 - \cos^k(\beta)) + c_b \cos^k(\beta) \quad [2],$$

где  $c_a$  — рассеянный цвет;  $c_d$  — цвет материала;  $c_b$  — цвет источника света;  $\alpha$  — угол между нормалью  $n$  и направлением на источник света  $s$ ;  $\beta$  — угол между нормалью  $n$  и биссектрисой  $\frac{s+v}{2}$  ( $v$  — направление на наблюдателя).

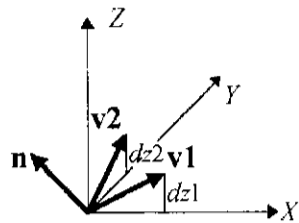


Рис. 9

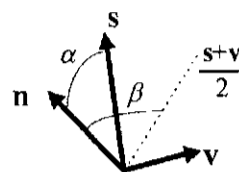


Рис. 10

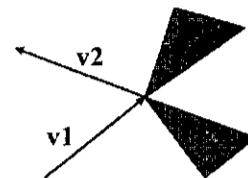


Рис. 11

Для конкретных значений  $c_a, c_d, c_b, s$  и  $v$  можно построить таблицу определения цвета по углу  $\varphi$ , что обеспечит высокую скорость построения изображения.

**Построение маски расстояний до символа.** Простейший вариант определения абсолютного значения и знака минимального расстояния от каждого элемента маски до контура символа — перебор всех векторов контура для каждой точки маски. Пусть размер маски  $N \times M$ , а  $L$  — число векторов в символе. Данный способ требует вычисления расстояния от точки до вектора  $N \times M \times L$  раз. Характерное время вычисления маски символа размером 72 пункта\* на компьютере с процессором 486DX2/66 МГц составляет секунды и десятки секунд, что нецелесообразно для интерактивной работы. Кроме того, на острых углах возникают ошибки при определении знака расстояния (внутри или снаружи символа находится пиксел). Из рис. 11 видно, что точки в выделенных областях располагаются слева от одного и справа от другого вектора. Для корректного определения знака требуются дополнительные вычисления.

Для быстрого вычисления маски нами был разработан волновой алгоритм определения минимального расстояния до контура, названный так по аналогии с известной моделью распространения волн в пространстве, в которой точки волнового фронта являются источниками вторичных волн. На очередном этапе список точек следующего фронта волны формируется из точек, прилегающих к точкам текущего фронта волны, при этом для каждой точки указывается ближайший вектор. Точки, уже обработанные или удаленные от контура на расстояние больше максимального, не рассматриваются. Первоначальный список точек фронта волны формируется из точек, непосредственно прилегающих к контуру.

На рис. 12 изображены точки вблизи формируемого фронта волны. Точки  $T_4, T_5, T_6$  уже обработаны, и для них вычислены ближайшие векторы контура, а также расстояние до этих векторов. Точки  $T_1, T_2, T_3$  принадлежат активному фронту волны, а  $T_0$  — формируемому фронту волны. Рассмотрим точку  $T_0$ . В нее придут вторичные волны из трех точек текущего фронта. Волна из точки  $T_1$  принесет с собой вектор  $V_1$ . Волна из точки  $T_2$  не изменит этого вектора. Волна из точки  $T_3$  может изменить его, если расстояние от точки  $T_0$  до вектора  $V_2$  меньше, чем расстояние до вектора  $V_1$ . Как видно из рисунка, в среднем в каждой точке будет вычисляться расстояние не более чем до двух векторов.

На каждом шаге расстояние от точек текущего фронта волны до контура увеличивается. Когда оно достигает максимального значения, алгоритм завершает свою работу.

**Алгоритм определения знака и абсолютного расстояния от точки до вектора.** Пусть нам дан вектор  $AB$  и точка  $C$ . Чему равно

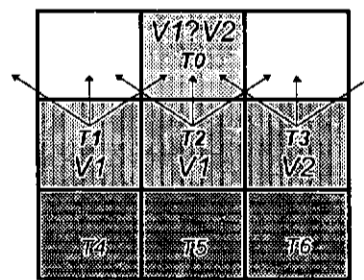


Рис. 12

\* 1 пункт = 1/72 дюйма. Пространственное разрешение VGA принято считать равным 96 точек на дюйм; таким образом, высота символа размером 72 пункта равна 96 точкам.

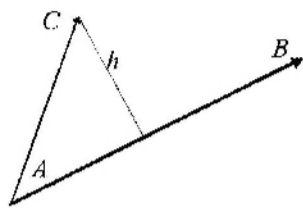


Рис. 13

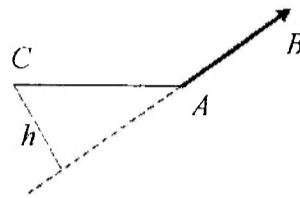


Рис. 14

расстояние от  $C$  до  $AB$ ? Опустим из точки  $C$  перпендикуляр  $h$  на  $AB$ . Если перпендикуляр  $h$  пересекает  $AB$  (рис. 13), то расстояние от точки  $C$  до вектора равно длине перпендикуляра  $h$ :  $L = \frac{|AB \times AC|}{|AC|}$ .

Если же перпендикуляр  $h$  не пересекает  $AB$  (рис. 14), то расстояние от точки до вектора равно расстоянию от  $C$  до ближайшей вершины  $AB$ .

Как правило, при построении изображения снаружи символа углы сглаживаются (рис. 15,  $b$ ), а внутри символа — сохраняются (рис. 15,  $a$ ), т. е. расстояние до вектора для точек внутри символа должно вычисляться по другой метрике. В случае когда перпендикуляр  $h$  выходит за пределы вектора (см. рис. 14), расстоянием до контура является длина самого длинного из двух перпендикуляров  $h_1$  и  $h_2$ , опущенных из точки  $C$  на векторы контура, прилегающие к ближайшей вершине вектора  $AB$  (рис. 16).

На рис. 17 показаны два изображения буквы  $B$ . Левое изображение построено с использованием одной (первой) метрики (углы внутри символа сглаживаются), правое — с использованием двух метрик (углы внутри символа сохраняются). Из рисунка видно, что изображение, полученное с использованием разных метрик, выглядит привычней.

Волновые алгоритмы определения расстояния до контура используют разные метрики, поэтому производится предварительное разделение маски на подобласти, лежащие внутри и снаружи символа. Начальный фронт волны задается отдельно для каждой из подобластей.

**Задание начального фронта волн.** При задании начального фронта обеих волн обрабатываются все пиксели, граничащие с контуром (рис. 18). Точки внутри контура (слева от вектора) изображены черным цветом, а точки вне контура (справа) — серым. Проблема с неопределенностью знака (см. рис. 14) может возникать только для пикселей вокруг вершин векторов (отмечены белым цветом). Для решения данной проблемы используется следующий алгоритм:

```
{обрабатывается пиксел  $x$  (рис. 18)}
if ( $x$  слева от  $AB$ ) AND ( $x$  слева от  $CA$ ) then ( $x$  внутри контура)
else if ( $x$  справа от  $AB$ ) AND ( $x$  справа от  $CA$ ) then ( $x$  снаружи контура)
else if ( $B$  слева от  $CA$ ) then ( $x$  снаружи контура)
else ( $x$  внутри контура)
```

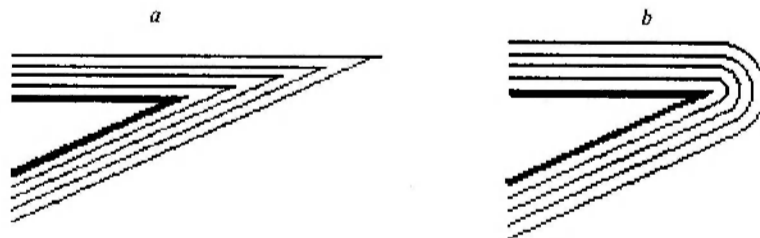


Рис. 15

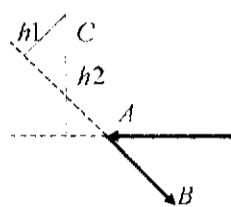


Рис. 16



Рис. 17

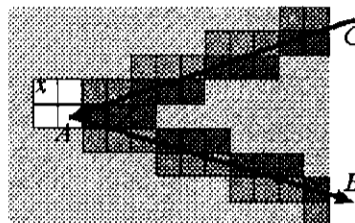


Рис. 18

После определения минимального расстояния во всех пикселах, прилегающих к контуру, маска расстояний оказывается разбитой на несколько четырехсвязных областей, ограниченных пикселами с одинаковым знаком (светло-серые области). Знак минимального расстояния внутри области проставляется в соответствии со знаком на границе области, для чего можно воспользоваться алгоритмом построчного сканирования маски. Пиксеты на границах маски по определению считаются наружными по отношению к контуру символа.

По окончании задания начальных фронтов волн для каждого фронта запускается волновой алгоритм вычисления расстояний до контура. Причем для волны, идущей внутрь символа, используется метрика с сохранением углов, а для волны, идущей наружу символа, — метрика со сглаживанием углов. В результате получаем готовую маску расстояний.

**Сравнение времен вычисления маски расстояний.** В таблице показаны средние времена вычисления маски символов размером 72 пункта на компьютере с процессором 486DX2/66 МГц. Справа от символа в первой строке указано время генерации маски с помощью волнового алгоритма, во второй строке — с помощью простого алгоритма (перебора), а в третьей строке показано отношение времен. Все времена даны в миллисекундах.

Из таблицы видно, что по сравнению с простейшим вариантом данный алгоритм обеспечивает скорость в 10—20 раз большую. Особенно значительный выигрыш в скорости получается на сложных символах, так как сложность вычисления маски с помощью волнового алгоритма практически не зависит от числа векторов. Для каждого символа маска вычисляется один раз в момент его первого появления на экране.

**Заключение.** Алгоритмы построения изображения по готовой маске работают настолько быстро, что построение изображения происходит практически незаметно для пользователя даже при одновременном использовании нескольких сложных трехмерных эффектов. Таким образом, использование данных алгоритмов позволяет сделать качественный переход в процессе подготовки титров на базе персонального компьютера. Интерактивный режим

Шрифт	"А"	"В"	"Я"	"Ф"	"Г"
Arial	377 1069 2.8	463 4873 10.5	465 2770 6	519 6663 12.8	455 4004 8.8
Courier New	349 3461 9.9	380 3459 9.1	406 5455 13.4	420 5583 13.3	362 4234 11.7
Times New Roman	107 647 6	123 986 8	98 639 6.5	121 1192 9.8	116 930 8
Windings	815 29507 36.2	900 29773 33	1592 75010 47	1189 42676 36	654 21799 33

работы, когда за доли секунды все изображение изменяется вслед за действиями пользователя, значительно ускоряет и облегчает процесс создания титров. Приложение, построенное на базе вышеописанных алгоритмов, обеспечивает скорость построения титров в несколько раз выше, чем наиболее распространенные программные знакогенераторы подобного класса.

#### СПИСОК ЛИТЕРАТУРЫ

1. Пугин Д. Телевизионные знакогенераторы // 625. 1996. № 2.
2. Magnenat-Thalmann N., Thalmann D. Computer Animation. N. Y.: Springer-Verlag, 1985.
3. Turkowski K. Anti-aliasing through the use of coordinate transformation // ACM Trans. on Graph. 1982. N 1 (3).

*Поступила в редакцию 11 июля 1996 г.*

---

---

Реклама продукции в нашем журнале — залог Вашего успеха!