

**В. В. Войтко, к. т. н., доц.; О. В. Романюк**

## **АНАЛИЗ МЕТОДОВ НОРМАЛИЗАЦИИ ВЕКТОРОВ НОРМАЛЕЙ ДЛЯ ЗАДАЧ ФОРМИРОВАНИЯ ТРЕХМЕРНЫХ ИЗОБРАЖЕНИЙ**

*Осуществлен анализ эффективных методов нормализации векторов нормалей для задач компьютерной графики. Приведены основные методы определения векторов нормалей для поверхностей, которые заданы аналитически и в виде полигональных данных.*

**Ключевые слова:** вектор нормали, нормализация векторов нормали, сферически-угловая интерполяция.

### **Введение**

Для достижения фотореалистичности в компьютерной графике необходимо точно воссоздать свойства поверхности и физически правильно описать эффекты освещения на сцене. Так одной из базовых операций, которая позволяет осуществить это, есть нахождение векторов нормали. Расчет векторов нормали – это неотъемлемая составная большинства методов формирования шероховатых и рельефных поверхностей. В методе бамп-меппинга [1], например, векторы нормали рассчитываются для вычисления попиксельного освещения поверхности. Степень освещения точки зависит от угла между нормалью и лучом света: чем меньше угол, тем больше освещенность в данной точке поверхности. Для неровных поверхностей нормали в каждой точке будут разными. Именно этот принцип является основой метода бамп-меппинга. А метод нормал-меппинг базируется на использовании карт нормалей. Цвет изображения говорит о том как сориентирована нормаль поверхности в данном пикселе изображения. Фактически карта нормали задает геометрию поверхности виртуальным обращением направления нормали в данном пикселе. В отличие от карт неровностей (бамп-карт) эффект карт нормалей хорошо видно даже на не очень блестящих материалах. Что касается методов формирования шероховатых поверхностей, которые используют карты смещения [2], то они также требуют просчета нормалей, поскольку задают смещение точек поверхности вдоль нормали.

Векторами нормалей задаются позиции наблюдателя и источника света. А также вектор нормали к поверхности задает ее локальную кривизну. Так, при закрашивании методом Фонга вдоль строки сканирования интерполируются значения векторов нормалей, которые потом используются в модели освещения Фонга (которая все еще остается очень популярной и широко используется в компьютерной графике) для вычисления диффузной и specular составляющей освещенности точки поверхности. При этом достигается лучшая локальная аппроксимация кривизны поверхности и, соответственно, получается более реалистичное изображение.

Согласно формуле закрашивания необходима нормализация векторов нормали. Нормализация вектора нормали [3] нуждается в выполнении трех операций деления, трех операций умножения, двух операций сложения и операции нахождения квадратного корня. В связи с этим можно констатировать, что векторные операции занимают весомую часть вычислительного процесса. Таким образом актуальным вопросом является упрощение процедуры нормализации с целью ее аппаратной реализации.

### **Цель исследования**

Целью данного исследования есть анализ методов нормализации векторов нормалей для использования в задачах компьютерной графики.

### Вектор нормали. Нахождение вектора нормали

Вектор нормали, или нормаль к плоской поверхности – это вектор, перпендикулярный к этой поверхности. Нормаль к неплоской поверхности в некоторой точке Р является вектором, перпендикулярным к касательной плоскости к этой поверхности в точке Р (см. рис. 1). Направление нормали определяет ориентацию поверхности в пространстве. В компьютерной графике вектор нормали используют для имитации геометрических деталей на плоских поверхностях. В данном случае функция будет определять малейшее отклонение от действительного направления нормали в каждой точке поверхности с целью создания отблесков и затененных участков.

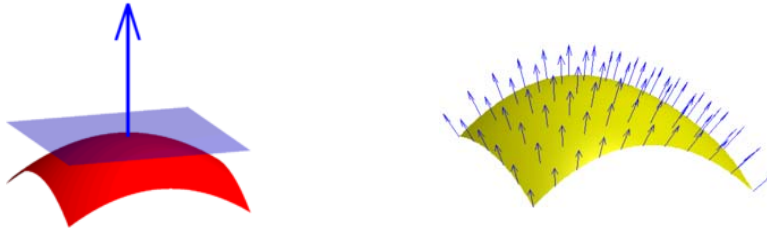


Рис. 1. Определение вектора нормали для неплоских поверхностей

На идеальной сфере, например, нормаль к точке поверхности имеет то же направление, что и вектор из центра сферы в эту точку. Для других типов поверхностей существуют другие лучшие способы нахождения нормалей, которые зависят от того, каким образом задается поверхность.

Следует помнить, что гладкие поверхности аппроксимируются большим количеством небольших плоских полигонов. Если векторы, перпендикулярные к этим полигонам используются в качестве нормалей аппроксимированных поверхностей, то сами поверхности выглядят сегментированными, поскольку пространство векторов не является непрерывным за пределами полигонов. Однако во многих случаях для модели существует точное математическое описание и в каждой точке может быть вычислен вектор истинной нормали. Использование истинных нормалей существенно улучшает результат визуализации (рис. 2).

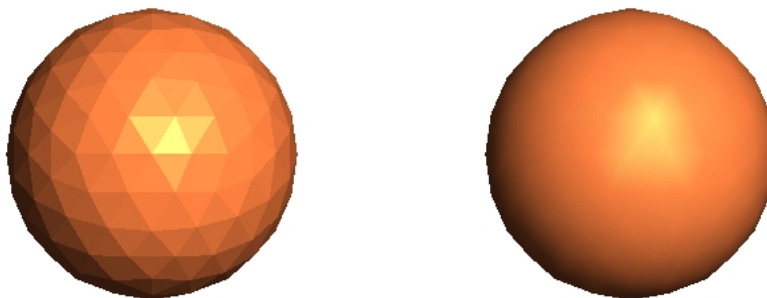


Рис. 2. Истинные нормали (справа) против полигональных нормалей (слева)

#### *Расчёт нормалей для аналитических поверхностей*

Аналитические поверхности – это плавные поверхности, которые описываются математическим уравнением (или некоторым набором уравнений). Во многих случаях, нормали проще находить для аналитических поверхностей, для которых имеется исчерпывающее описание в такой форме:

$$V(s, t) = [X(s, t) \ Y(s, t) \ Z(s, t)],$$

где  $s$  и  $t$  определенные в некотором пространстве, а  $X$ ,  $Y$  и  $Z$  – дифференцируемые

функции двух переменных. Для того, чтобы найти нормаль, необходимо вычислить частичные производные  $\frac{\partial V}{\partial s}$  и  $\frac{\partial V}{\partial t}$ , которые являются векторами, касательными к поверхности в направлениях  $s$  и  $t$ . Их векторное произведение  $\frac{\partial V}{\partial s} \times \frac{\partial V}{\partial t}$  перпендикулярное к ним обоим и, как следствие, перпендикулярное к поверхности. Следующая формула отображает процесс вычисления векторного произведения двух векторов:

$$[v_x v_y v_z] \times [w_x w_y w_z] = [(v_y w_z - w_y v_z)(v_z w_x - w_z v_x)(v_x w_y - w_x v_y)]. \quad (1)$$

Для нормализации результирующего вектора  $[x, y, z]$  необходимо найти его длину:

$$l = \sqrt{x^2 + y^2 + z^2} \quad (2)$$

и разделить на нее каждый из компонентов вектора.

В качестве примера для таких расчетов возьмем аналитическую поверхность  $V(s, t) = [2s \quad t^2 \quad 4 + 2st]$ . Рассчитаем частичные производные и с помощью формулы (1) найдем векторное произведение полученных векторов:

$$\frac{\partial V}{\partial s} = [2 \quad 0 \quad 2t], \quad \frac{\partial V}{\partial t} = [0 \quad 2t \quad 2s], \quad \frac{\partial V}{\partial s} \times \frac{\partial V}{\partial t} = [-4t^2 \quad -4s \quad 4t].$$

Таким образом, если в данном примере  $s=2$  и  $t=1$ , соответствующей точкой поверхности является точка с координатами  $(4, 1, 8)$ , а вектором нормали будет вектор  $(-4, -8, 4)$ . Длина этого вектора равняется 9.8, тогда вектор нормали единичной длины равняется  $(-4/9.8, -8/9.8, 4/9.8) = (-0.408163, -0.816326, 0.408163)$ .

*Расчёт нормалей по полигональным данным*

Очень часто поверхности заданы в виде полигональных данных. Для того, чтобы они выглядели гладкими, а не сегментированными, нужно вычислять нормали к поверхности. В большинстве случаев простейшим способом сделать это есть вычисление вектора нормали для каждого полигонального фрагмента поверхности, а потом усреднить нормали соседних фрагментов. На рис. 3 изображена поверхность и ее полигональная аппроксимация.

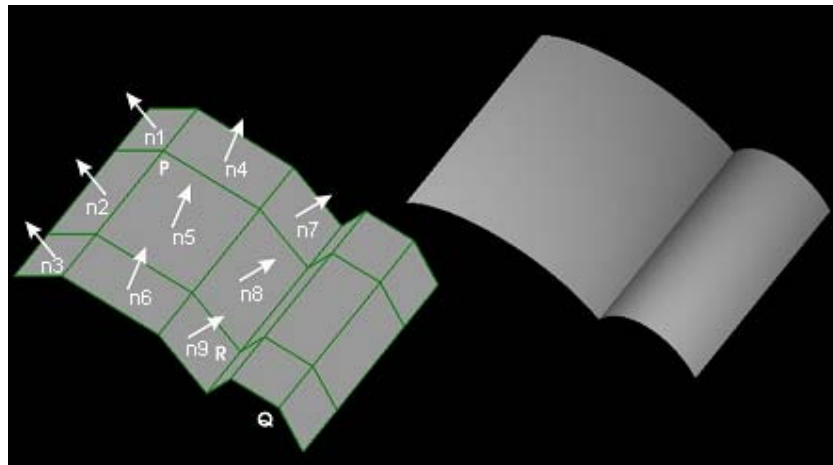


Рис. 3. Усреднение векторов нормали

Для того, чтобы найти вектор нормали для плоского полигона нужны три любые вершины полигона  $v_1, v_2$  и  $v_3$ , которые не лежат на одной прямой. Векторное произведение

$$[v_1 - v_2] \times [v_2 - v_3] \quad (3)$$

и будет нормалью к полигону. Как правило, результирующий вектор нужно

нормализовать. Потом нужно усреднить нормали соседних полигонов, чтобы не предоставлять больше веса одному из них. Например, если в случае, показанному на рис. 3,  $n_1, n_2, n_4$  и  $n_5$  – нормали полигонов, которые соединяются в точке Р, то нужно найти сумму векторов  $n_1+n_2+n_4+n_5$ , а потом нормализовать полученный вектор. Результирующий вектор может служить нормалью в точке Р.

В некоторых моделях бывают плавные части и острые углы (точка R на рис. 3 размещена на таком ребре). В таком случае не нужно усреднять нормали соседних полигонов, а наоборот – полигоны с одной стороны ребра должны быть нарисованы с использованием одной нормали, а со второй стороны – с использованием другой нормали.

### Анализ методов нормализации векторов нормали

Научными работниками фирмы VIDIA была получена формула [4] для приближенной нормализации векторов нормалей с использованием одного шага итерации Ньютона – Рафсона:

$$\vec{N}_n = \frac{\vec{N}}{3}(3 - \vec{N} \cdot \vec{N}). \quad (4)$$

Расчет согласно приведенной формулы требует одной операции деления, 6 операций умножения и 3 операций сложения. Большие абсолютные погрешности определения ортогональных составных вектора ограничивают использование формулы для задач закрашивания.

Согласно подходу Р. Лиона выражение  $1/\sqrt{\vec{N} \cdot \vec{N}}$ , которое используется для нормализации вектора нормали  $\vec{N}$ , раскладывают в ряд Тейлора и ограничиваются первыми тремя членами. Тогда [5]:

$$\vec{N}_n = \vec{N} \left( 1 - \frac{1}{2}((\vec{N} \cdot \vec{N}) - 1) + \frac{3}{8}((\vec{N} \cdot \vec{N}) - 1)^2 \right). \quad (5)$$

Формула хотя и пригодная для аппаратной реализации, но не обеспечивает приемлемой точности.

Интерполяцию единичных векторов нормалей между начальным  $\vec{N}_a$  и конечным  $\vec{N}_b$  векторами, которые имеют единичную длину, можно выполнить за формулой [6]:

$$\vec{N}(w) = \vec{N}_a \frac{\sin((1-w)\psi)}{\sin \psi} + \vec{N}_b \frac{\sin(w\psi)}{\sin \psi}, \quad (6)$$

где  $w \in [0, 1]$ , а  $\psi$  – угол между векторами нормалей  $\vec{N}_a$  и  $\vec{N}_b$ .

Однако, существенным недостатком метода является необходимость расчета тригонометрических функций  $\sin$ ,  $\arccos$  как для нахождения конкретного вектора нормали, так и для расчета неизвестных параметров  $w$  и  $\psi$ . Кроме того, расчет  $\vec{N}(w)$  предусматривает выполнение в цикле нормализации операции деления.

В работе [7] предложено использовать принцип дихотомии для проведения нормализации векторов нормалей путем последовательного деления пополам угла между векторами нормалей в начальной и конечной точках строки растеризации треугольника. Общая формула имеет вид:

$$\vec{N}_{\left(\frac{1}{2^{n+1}}\right)} = \frac{\vec{N}_a + \vec{N}_{\frac{1}{2^n}}}{\sqrt{2 + z_{2^n}}}, \quad (7)$$

где  $\vec{N}_a$  – начальный вектор; обозначение  $\left(\frac{1}{2^n}\right)$  означает количество сегментов, которое

получено на  $n$ -итерации при последовательном делении пополам сегментов строки растеризации треугольника при условии, что при  $n = 0$  сегмент равняется строке растеризации;  $z_{2^n} = \sqrt{2(1 + \cos \frac{\psi}{2^{n-1}})}$ ;  $\psi$  – угол между векторами нормалей  $\vec{N}_a$  (начальным) и  $\vec{N}_b$  (конечным).

На каждой итерации для нахождения знаменателя выполняется лишь одна операция сложения и одна операция определения квадратного корня.

Преимуществом данного метода является то, что выражение  $\frac{1}{\sqrt{2+z_{2^n}}}$  можно аппроксимировать рядом Чебышева, что обеспечивает незначительные погрешности аппроксимации. При использовании многочлена первой степени

$$\frac{1}{\sqrt{2+z_{2^n}}} \approx -0,07 \cdot z_{2^n} + 0,64 \quad (8)$$

максимальная абсолютная погрешность аппроксимации не превышает 0,0005, а относительная 0,12 %. Данную формулу целесообразно использовать для экранов с небольшой раздельной способностью, для которых треугольники, которые составляют поверхность трехмерного объекта, имеют незначительные размеры.

При использовании многочлена второй степени

$$\frac{1}{\sqrt{2+z_{2^n}}} \approx 0,014 \cdot z_{2^n}^2 - 0,119 \cdot z_{2^n} + 0,681 \quad (9)$$

максимальная абсолютная погрешность аппроксимации не превышает  $2 \cdot 10^{-5}$ , а относительная 0,004 %. Анализ показал, что при использовании этой аппроксимационной формулы время расчета вектора  $\vec{N}_{(\frac{1}{2^n})}$  уменьшается в 2,5 раза сравнительно с классической реализацией.

Другой метод нормализации векторов нормали, предложенный в работе [8], предусматривает использование квадратичной интерполяции при условии, что известны единичные векторы нормалей в начальной и конечной точках  $i$ -той строки растеризации треугольника. Промежуточные значения векторов нормалей в строке растеризации треугольника находятся по формуле:

$$\vec{N}_{i,t} = \vec{G}_i \cdot t^2 + \vec{P}_i \cdot t + \vec{Q}_i. \quad (10)$$

Пусть  $\vec{N}_{i,l}$ ,  $\vec{N}_{i,p}$ ,  $\vec{N}_{i,c}$  – соответственно векторы нормалей в левой, правой и средней точках строки растеризации треугольника. При  $t = 0$   $\vec{N}_{i,l} = \vec{Q}_i$ . В правой точке строки растеризации  $t = 1$ , поэтому  $\vec{N}_{i,p} = \vec{G}_i + \vec{P}_i + \vec{Q}_i$ . Поскольку в средней точке строки растеризации  $t = 1/2$ , то  $\vec{N}_{i,c} = \frac{\vec{G}_i}{4} + \frac{\vec{P}_i}{2} + \vec{Q}_i$ . После ряда преобразований можно найти следующее выражение для нахождения вектора нормали в средней точке:

$$\vec{N}_{i,c} = \frac{1}{\sqrt{2(1 + \vec{N}_{i,l} \cdot \vec{N}_{i,p})}} (\vec{N}_{i,l} + \vec{N}_{i,p}). \quad (11)$$

Преимуществом данного метода является возможность аппроксимировать выражение  $\frac{1}{\sqrt{2(1 + \vec{N}_{i,l} \cdot \vec{N}_{i,p})}}$  полиномом Чебышева второй степени, что разрешает уменьшить время

расчета вектора нормали в средней точке строки растеризации более как в 2,5 раза. Анализ показал, что при программной реализации предложенного метода время расчета векторов нормалей для среднего треугольника уменьшилось в 2,8 раза сравнительно с классической реализацией.

Для формирования векторов нормали можно использовать и сферически-угловую интерполяцию [9] (рис. 4). Данный метод целесообразно использовать при условии оперирования с углами над векторами нормалей. Промежуточные значения векторов нормали за этим методом определяются по формуле:

$$\vec{N}(t) = \vec{N}_a \cdot \cos(t \cdot \varphi) + \vec{N}_k \cdot \sin(t \cdot \varphi), \quad (12)$$

где  $t$  – номер пикселя вдоль строки растеризации,  $t \in [0, l]$ ;  $\vec{N}_a$  – начальный вектор;  $\vec{N}_k = \frac{\vec{N}_b - \vec{N}_a (\vec{N}_b \cdot \vec{N}_a)}{\sqrt{1 - (\vec{N}_b \cdot \vec{N}_a)^2}}$  – нормализованный вектор нормали;  $\vec{N}_b$  – конечный вектор;  $\varphi = \psi / m$  – угол между двумя соседними векторами нормалей, где  $\psi = \arccos(\vec{N}_a \cdot \vec{N}_b)$  и  $m$  – длина строки растеризации.

Данную формулу после ряда преобразований можно записать в итерационной форме:

$$\vec{N}(t+1) = 2\vec{N}(t) \cdot \cos \varphi - \vec{N}(t-1). \quad (13)$$

Из последней формулы можно сделать вывод, что единичный вектор нормали при сферически-угловой интерполяции можно найти через два предыдущих значения.

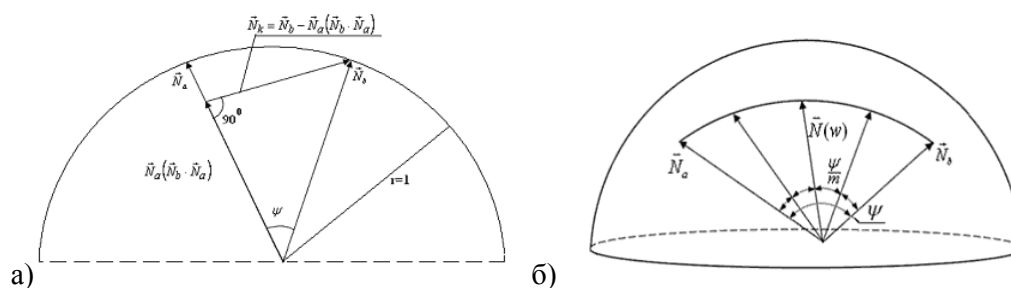


Рис. 4. Определение вектора  $\vec{N}_k$  (а) и сферически-угловая интерполяция векторов нормали (б)

Анализ показал, что при программной реализации согласно формулы (13) время расчета векторов нормалей среднего треугольника уменьшилось в 1,7 раз сравнительно с вычислением единичных векторов нормалей путем угловой интерполяции. Таким образом, можно констатировать, что достигается существенное повышение быстродействия закрашивания поверхности.

### Выводы

Нахождение векторов нормали и осуществление их нормализации занимает весомую часть процесса закрашивания трехмерных изображений в задачах компьютерной графики. Большое количество научных работников посвятили свои исследования упрощению процедуры нормализации векторов нормалей. Анализ, проведенный в данной работе, показал, что на сегодня наиболее эффективными методами нормализации являются методы, которые используют квадратичную интерполяцию и сферически-угловую интерполяцию при условии работы с углами над векторами нормалей, а также принцип дихотомии. Данные методы разрешают улучшить время расчета векторов нормалей от 1,7 до 2,8 раз в сравнении с классической нормализацией, что в свою очередь существенно повышает быстродействие закрашивания трехмерных изображений.

## СПИСОК ЛИТЕРАТУРЫ

1. Blinn J. F. Simulation of wrinkled surfaces. // Computer Graphics (SIGGRAPH '78 Proceedings). – 1978. – № 12. – P. 286 – 292.
2. Donnelly W. Per-pixel displacement mapping with distance functions // In GPU Gems II, M. P., (Ed.). Addison-Wesley. – 2005. – P. 123 – 136.
3. Романюк О. Н. Новий підхід до підвищення реалістичності зафарбовування тривимірних об'єктів за методом Гуро // Інформаційні технології та комп'ютерна інженерія. — 2005. — №2. — С. 106 — 109.
4. Barrera T., Hast A., Bengtsson E. Fast Near Phong-Quality Software // Shading SCCG'06 full paper. – 2006. – P. 44 – 50.
5. Lyon R. F. Phong Shading Reformulation for Hardware Renderer Simplification // Apple Technical Report. — 1993. — № 43. — 34 p.
6. Hast A. Improved Algorithms for Fast Shading and Lighting // Acta Universitatis Upsaliensis. Uppsala. – 2004. – 116 p.
7. Романюк О. Н. Комбіноване використання бінарної та кодової лінійної інтерполяції для нормалізації векторів нормалей при зафарбовуванні тривимірних об'єктів // Вестник Херсонського національного технічного університета. — 2006. — № 25. — С. 408 — 411.
8. Романюк О. Н. Використання квадратичної інтерполяції для зафарбовування тривимірних графічних об'єктів // Реєстрація, зберігання і обробка даних. — 2006. — Т. 8. — № 4. — С. 31 — 37.
9. Романюк О. Н. Реалізація рендерингу Фонга з використанням сферично-кутової інтерполяції / О. Н. Романюк, А. В. Чорний // Вісник Вінницького політехнічного інституту. — 2004. — № 3. — С. 66 – 71.

**Войтко Виктория Владимировна** – к. т. н., доцент кафедри програмного забезпечення; тел. 8 (0432) 59-84-83.

**Романюк Оксана Владимировна** – аспірант кафедри програмного забезпечення інституту інформаційних технологій і комп'ютерної інженерії; тел. 8 (0432) 65-85-06. Вінницький національний технічний університет.