НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Системы компьютерной обработки изображений

Лабораторная работа № 2

«Вычисление яркости в точках на поверхности»

Выполнил студент:

Хоанг Ван Куан

Группа № P3366

Преподаватель:

Жданов Дмитрий Дмитриевич

г. Санкт-Петербург 2025

## Цель

## Изучить, как вычисляется яркость точках на плоскости треугольника с учетом освещения от точечных источников света и диффузно отражательных свойств поверхности.

## Используемые формулы

1. «Цветная» интенсивность источника под углом к оси источника света:

*, где – «цветная» интенсивность источника света в направлении его оси , θ - угол между направлением распространения света и осью источника света, – диаграмма излучения.*

1. «Цветная» освещенность точки:

, *где – «цветная» интенсивность света, α - угол между направлением света и нормалью к освещаемой поверхности, R - расстояние от источника света до рассматриваемой точки.*

1. Перевод локальных координат точки в плоскости в глобальные

*, где и смещения по ребрам треугольника.*

1. Вычисление вектора нормали плоскости треугольника через 3 точки:

1. Вектор от точки плоскости до источника света:

1. Яркость точки:

, *где - освещённость от i-го источника света, - координаты точки, - направление на наблюдения, - направление на источник света, - двунаправленная функция отражения (BRDF), – «цвет» поверхности.*

1. Двунаправленная функция отражения – как отражается или поглощается поверхностью в зависимости от разных углов падения

, *где*  *- коэффициент диффузного отражения поверхности, - коэффициент «зеркальности» поверхности, – коэффициент, определяющий ширину блика, - средний вектор между направлениями освещения и наблюдения, - нормаль к поверхности в точке освещения.*

1. Средний вектор

.

A diagram of a triangle with arrows and lines

AI-generated content may be incorrect.

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

## Шаги алгоритма

## Вычислите глобальные координаты PT и вектор нормали плоскости N

## Для каждого источника света:

## Рассчитайте освещенность ​

## Вычислитель векторного среднего значения

## Рассчитайте яркость по формуле

## Программа для расчета яркости

public Vector illumination(Vector N, Vector s, Vector O, Vector I0\_RGB) throws Exception {  
 double R2 = s.dot\_product(s);  
 double cos\_alpha = s.cos(N);  
 double cos\_theta = s.cos(O);  
  
 if (cos\_alpha < 0 || cos\_theta < 0) return new Vector(new double[]{0, 0, 0});  
 return I0\_RGB.scale\_multiply(cos\_alpha \* cos\_theta / R2);  
}

// Ярко  
public Vector brightness(Point P0, Point P1, Point P2, Point PL1, Point PL2, Vector O1, Vector O2,  
 Vector I01\_RGB, Vector I02\_RGB, double x, double y,  
 Point V, Vector K\_RGB, double kd, double ks, double ke) throws Exception {  
 Vector N = normal(P0, P1, P2);  
 Point PT = PT(P0, P1, P2, x, y);  
 Vector v = new Vector(V, PT);  
 if (v.dot\_product(N) < 0) N = N.scale\_multiply(-1);  
 Vector s1 = new Vector(PL1, PT);  
 Vector s2 = new Vector(PL2, PT);  
  
 Vector E1\_RGB = illumination(N, s1, O1, I01\_RGB);  
 Vector E2\_RGB = illumination(N, s2, O2, I02\_RGB);  
  
 Vector h1 = s1.Average(v);  
 Vector h2 = s2.Average(v);  
 double coeff1 = kd + ks \* Math.*pow*(Math.*max*(0, h1.dot\_product(N)), ke);  
 double coeff2 = kd + ks \* Math.*pow*(Math.*max*(0, h2.dot\_product(N)), ke);  
  
 Vector term1 = E1\_RGB.scale\_multiply(coeff1).scale\_multiply(K\_RGB.dot\_product(new Vector(new double[]{1, 1, 1})));  
 Vector term2 = E2\_RGB.scale\_multiply(coeff2).scale\_multiply(K\_RGB.dot\_product(new Vector(new double[]{1, 1, 1})));  
 return term1.plus(term2).scale\_multiply(1 / Math.*PI*);  
}

**Пример входных данных и результаты**







Вычисленные значения освещенности для точек, заданных локальными координатами:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Вычисленные значения освещенности для тех же точек, заданных локальными координатами:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Вычисленные значения яркостей для тех же точек на плоскости с заданными условиями наблюдения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | (0.13, 0.13, 0.13) | (0.13, 0.13, 0.13) | (0.13, 0.13, 0.13) | (0.14, 0.14, 0.14) | (0.15, 0.15, 0.15) |
|  | (0.15, 0.15, 0.15) | (0.15, 0.15, 0.15) | (0.16, 0.16, 0.16) | (0.17, 0.17, 0.17) | (0.18, 0.18, 0.18) |
|  | (0.18, 0.18, 0.18) | (0.18, 0.18, 0.18) | (0.19, 0.19, 0.19) | (0.20, 0.20, 0.20) | (0.22, 0.22, 0.22) |
|  | (0.21, 0.21, 0.21) | (0.21, 0.21, 0.21) | (0.22, 0.22, 0.22) | (0.25, 0.25, 0.25) | (0.27, 0.27, 0.27) |
|  | (0.24, 0.24, 0.24) | (0.24, 0.24, 0.24) | (0.26, 0.26, 0.26) | (0.30, 0.30, 0.30) | (0.34, 0.34, 0.34) |

## Вывод

В данной работе был изучен и реализован алгоритм вычисления яркости точки на плоскости с учетом освещенности от нескольких источников света. Используя заданные входные данные, программа корректно вычисляет значения освещенности и яркости для различных точек плоскости, учитывая направленность света и свойства поверхности.