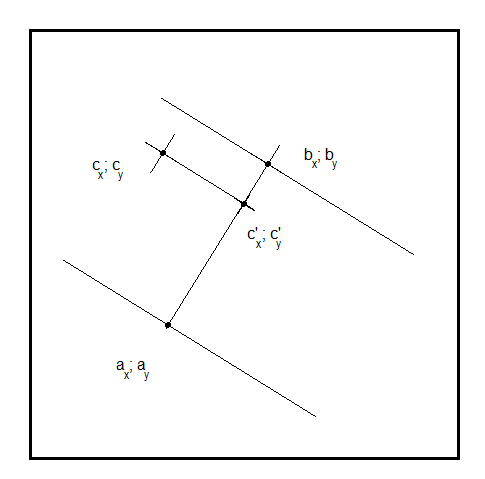
**Линейный градиент + линеарный градиент**

  
рис. 1 Модель рендера линеарного градиента

**Вычисление координаты c’:**

Точки a и b – начальная и конечная позиции градиента, c – текущая точка рендеринга, f – пересечение перпендикуляра к ab из c.

По точке f вычисляется локальная позиция текущей точки градиента. Локальная позиция – float значение, равное 0, если f = a, равно 1, если f = b. Если f > 1, или < 0, точка c закрашивается в значение одного из крайних цветов, иначе закрашивается в зависимости от локального положения.

Вычисляется f следующим образом:

Имеем уравнение прямой ab: , подставим ax, ay, bx, by, дабы найти k и b:

, откуда вычислим корни: Параметры k и p не зависят от точки c, а значит их можно вычислить заранее и передать в шейдер.

Имеем уравнение прямой, перпендикулярной к ab и проходящее через c:  
 , x и y в данном уравнении и есть координаты . Составим систему уравнений:

, её решением будет:

**Вычисление локальной позиции:**

Задача определить, насколько близко f подходит к b или a и с какой стороны, если f < 0, значит, эта точка ещё не подошла к а и ещё дальше находится от b, если > 0, но < 1: она находится между a и b, если f > 1, то точка находится за точкой b.

Решить можно 2 способами (так как плоскость это 2 измерения): значение локальной позиции *l*: , так как знаменатель одного из уравнений может обратиться в 0, это происходит, когда линии параллельны осям координат, поэтому вводим условие: ищем наибольший знаменатель, выполняем выражение с максимальным знаменателем, гарантируется, что оба они не будут равны 0.

**Вычисление цвета промежуточных значений *l*:**

Точки в шейдер передаются 3 массивами:

\_GradientPoints,   
 \_GradientPointsPositions,   
 \_GradientTypePoints.

1 хранит все данные точек,

2 ссылается на 1 массив, указывая начало данных для каждой точки в 1 массиве,

3 указывает тип точки, благодаря нему становится понятно, имеем ли мы дело с COLOR\_POINT, или же MIDDLE\_POINT.

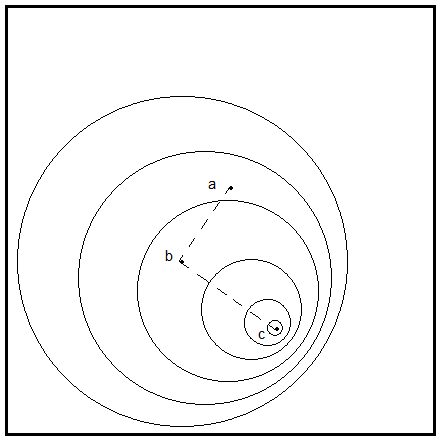
Точки в линейном градиенте бывают 2 видов:

COLOR\_POINT – задаёт цвет и позицию текущей точки на ab,

MIDDLE\_POINT – промежуточные точки нужны для кастомизации интерполяции цвета по пути от i COLOR\_POINT, до i+1. MIDDLE\_POINT может быть много, а может и не быть вообще. Если MIDDLE\_POINT на пути между 2 COLOR\_POINT отсутствуют – цвет линейно интерполируется между 2 COLOR\_POINT, от цвета i точки (значение интерполяции 0), до цвета i+1 (значение интерполяции 1). Если MIDDLE\_POINT на пути присутствуют в количестве n штук, происходит следующее: значение интерполяции в каждой j MIDDLE\_POINT точке равно , , между точками j и j+1 значения линейно интерполируются от значения интерполяции в j, до значения интерполяции в j+1. При этом друг относительно друга точки MIDDLE\_POINT могут быть случайно удалены друг от друга соблюдая порядок, например: ближе всего к i точке должна быть j=1, ближе всего к i+1, j=n.

Интерполяция цвета от i, до i+1 COLOR\_POINT выглядит следующим образом: с помощью \_GradientTypePoints и *l* находим i и i+1 точки COLOR\_POINT, затем находим j и j+1 точки MIDDLE\_POINT, если таковые имеются, интерполируем цвет между точками i и i+1, интерполируя позицию j и j+1.

**Радиальный градиент**

   
рис. 2 Модель рендера радиального градиента

Имеются 3 точки: а – центр изображения, ничего особенного, b – центр круга, передаётся в шейдр, c – смещение центра круга, локальная величина относительно самого круга, передаётся в шейдер, так же имеем r – радиус круга (когда b совпадает с a и r = 0.5 , квадрат описывает основной круг), r принимает значения от 0 до 0.5 .

В качестве точек радиальный градиент хранит те же самые точки линейного градиента и рендерятся они аналогично (см. предыдущий раздел), однако величина *l*, задающая значение интерполяции для цвета описывается иначе, чем в линеарном градиенте.

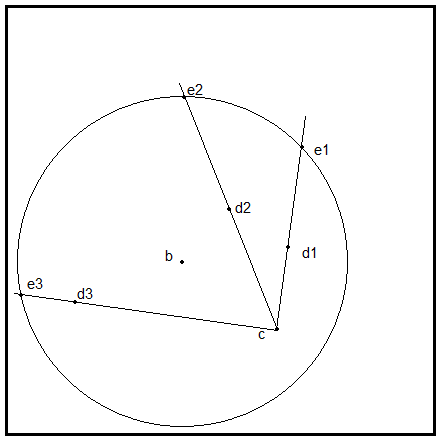


рис. 3 Визуализация точек d и e

За текущую точку обозначим точку d (на рисунке примеры: d1, d2, d3), тока e – будет пересечением прямой cd с основным кругом, , остаётся только найти координаты точки e.

Имеем уравнение окружности и уравнение прямой проходящей через 2 точки, составим из них систему:

Решение этой системы уравнений с 2 неизвестными x и y даст координаты точки e, но так как первое уравнение имеет 2 степень, корней получается 2, нам же нужен только 1 из них, так как второй корень даст координаты противоположной точки пересечения по другую сторону от c, чего нам не нужно.

Разделение на 2 корня появляется там где появляется дискриминант ( и ), экспериментальным путём было выяснено что нам нужен корень с , решим систему уравнений и найдём его: , где

Далее *l* вычисляем так же как вычисляли в линеарном градиенте.