Отчёт по задаче: Моделирование линий, плоскостей и оптических приборов
(интерпретация через трёхмерные прямые)

Автор: Иванов И.И.

17 апреля 2025 г.

1 Постановка задачи

Требуется изучить следующие аспекты:

1. **Задание прямой** в трёхмерном пространстве в параметрической форме:
$$\mathbf{v}(t) = \mathbf{P} + t\,\mathbf{d},$$
где $\mathbf{P} = (x_0, y_0, z_0)$ — точка на прямой, $\mathbf{d} = (n, m, p)$ — направляющий вектор.

2. **Пересечение двух прямых** в пространстве — решение системы
$$\begin{cases} x_{01} + t\,n_1 = x_{02} + s\,n_2, \\ y_{01} + t\,m_1 = y_{02} + s\,m_2, \\ z_{01} + t\,p_1 = z_{02} + s\,p_2. \end{cases}$$
Если решение существует (и прямые не параллельны/не скрещены), получаем точку пересечения $\mathbf{r}^\star$.

3. **Пересечение прямой с плоскостью**, заданной уравнением
$$A\,x + B\,y + C\,z + D = 0.$$
Подстановка параметрического уравнения прямой даёт значение t.

4. **Применение** полученных формул к простым оптическим системам (тонкая линза, микроскоп, телескоп) — путём «трассировки лучей» в приближении 2D/3D.

2 Теоретические основы

2.1 Параметрическая форма прямой

В трёхмерном пространстве $\mathbb{R}^3$ **прямая** может быть задана векторно:
$$\mathbf{r}(t) = \mathbf{P} + t\,\mathbf{d},$$
где $t \in \mathbb{R}$, $\mathbf{P} = (x_0, y_0, z_0)$, $\mathbf{d} = (d_x, d_y, d_z)$.

Скалярно (по координатам) это
$$\begin{cases} x(t) = x_0 + t\,d_x, \\ y(t) = y_0 + t\,d_y, \\ z(t) = z_0 + t\,d_z. \end{cases}$$

2.2 Пересечение прямых

Рассмотрим две прямые:

$$\mathbf{r}_1(t) = \mathbf{P}_1 + t\,\mathbf{d}_1, \quad \mathbf{r}_2(s) = \mathbf{P}_2 + s\,\mathbf{d}_2.$$

Пересечение означает, что существует $(t^\star, s^\star)$ такое, что $\mathbf{r}_1(t^\star) = \mathbf{r}_2(s^\star)$. В координатном виде:

$$\begin{cases} x_{01} + t\,d_{1x} = x_{02} + s\,d_{2x}, \\ y_{01} + t\,d_{1y} = y_{02} + s\,d_{2y}, \\ z_{01} + t\,d_{1z} = z_{02} + s\,d_{2z}. \end{cases}$$

Решая эту систему относительно (t, s), находим при наличии решения точку пересечения.

2.3 Пересечение прямой с плоскостью

Пусть плоскость задана уравнением

$$A\,x + B\,y + C\,z + D = 0.$$

Подставляя параметры прямой $\mathbf{r}(t) = \mathbf{p} + t\,\mathbf{d}$, получаем скалярное уравнение:

$$A(x_0 + t\,d_x) + B(y_0 + t\,d_y) + C(z_0 + t\,d_z) + D = 0,$$

$$t\,(A\,d_x + B\,d_y + C\,d_z) = -\big(A\,x_0 + B\,y_0 + C\,z_0 + D\big).$$

Если знаменатель $A\,d_x + B\,d_y + C\,d_z \neq 0$, то

$$t^\star = -\frac{A\,x_0 + B\,y_0 + C\,z_0 + D}{A\,d_x + B\,d_y + C\,d_z}.$$

Тогда точка пересечения: $\mathbf{r}(t^\star) = \mathbf{p} + t^\star\mathbf{d}$.

3 Численная реализация и ход лучей в модели «тонкой линзы»

В прилагаемом `Jupyter Notebook` (см. листинг или файл) мы реализовали классы:

- `Line` — хранит информацию о прямой (`offset`, `multiply`) и методы:
  - `get_dot(t)` — получение координат точки при параметре t;
  - `intersection(other)` — решение системы для пересечения с другой прямой.
- `Plane` — задаёт плоскость с параметрами (A, B, C, D) и метод `intersection_with_line(line)`, возвращающий точку пересечения.

Для **оптической** задачи (тонкая линза, микроскоп, телескоп):

1. **Тонкая линза** рассматривается как:
   - Луч 1: проходит через точку предмета и центр линзы (не преломляется).
   - Луч 2: параллелен оптической оси, затем проходит через фокус.

   Пересечение этих двух лучей даёт положение соответствующей точки изображения.
2. **Микроскоп, телескоп** строятся последовательно: изображение, сформированное первой линзой, становится «предметом» для второй и т.д. При этом требуется несколько раз вызвать функцию преобразования.

4 Основные фрагменты кода

Ниже для примера приводим несколько ключевых строк. Полный код см. в ноутбуке:

```
class Line:
    def __init__(self, x0, y0, z0, m, n, p):
        self.offset = np.array([x0, y0, z0], dtype=float)
        self.multiply = np.array([m, n, p], dtype=float)

    def get_dot(self, t):
        return self.offset + t * self.multiply

    def intersection(self, other):
        A = np.array([self.multiply, -other.multiply,
                      np.cross(self.multiply, other.multiply)]).T
        B = other.offset - self.offset
        try:
            t, s, _ = np.linalg.solve(A, B)
        except np.linalg.LinAlgError:
            return None
        return self.get_dot(t)

def thin_lens_image_input_lines(
    input_image, f, a
):
    #                          b = (a*f)/(a-f)
    #                            :                                        ,
    # ...
    #
    return output_image, b, M, error
```

5 Результаты

В ноутбуке продемонстрировано несколько сценариев:

5.1 1. Тонкая линза

Исходное изображение (рис. ??) проходит через лупу (тонкую линзу) при заданных f и a. Результат — увеличенное (или уменьшенное) изображение, которое может оказаться перевёрнутым, если $a > f$ (действительное изображение).

5.2 2. Микроскоп

Состоит из двух линз (объектив и окуляр), расположенных на некотором расстоянии. Пошагово получаем:

1. Промежуточное изображение, даваемое объективом,
2. Финальное увеличенное изображение после окуляра.

В ноутбуке выведены результаты в виде двух-трёх картинок.

5.3 3. Телескоп

По аналогии: также две линзы, но настраиваем их так, чтобы параллельные лучи на входе давали параллельные лучи на выходе. Угловое увеличение $\gamma = f_{\text{об}}/f_{\text{ок}}$.

6 Выводы

1. Реализован базовый инструментарий для определения **пересечения прямых и плоскостей** в 3D-пространстве.
2. Данный инструментарий применён для **трассировки лучей** в простейших оптических моделях (тонкая линза, микроскоп, телескоп).
3. Код позволяет "проецировать" цифровые изображения (набор точек) через модель оптической системы, получая новое (искажённое, увеличенное) изображение.
4. Подобный подход может быть расширен для более сложных систем, многократных отражений/преломлений и реальных (толстых) линз.

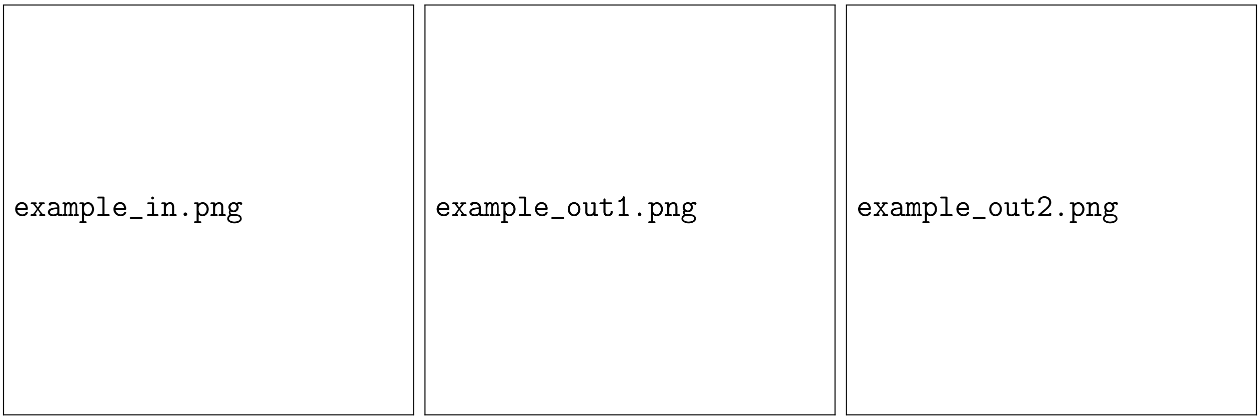

Рис. 1: Пример: исходное изображение (слева), промежуточное (по первой линзе) и финальное изображение (по второй линзе).