Параметризации поворотов. RANSAC

Сергей Кривохатский

СПбГУ, Современное программирование

7 октября 2022 г.

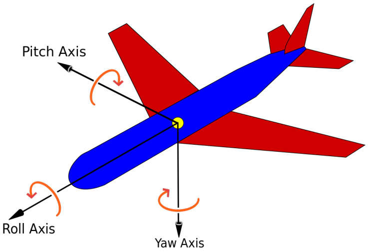
Углы Эйлера I

Три угла, задающих последовательное вращение вокруг заданных осей объекта

$$\boldsymbol{R}_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}$$

$$\boldsymbol{R}_y(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}$$

$$\boldsymbol{R}_z(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

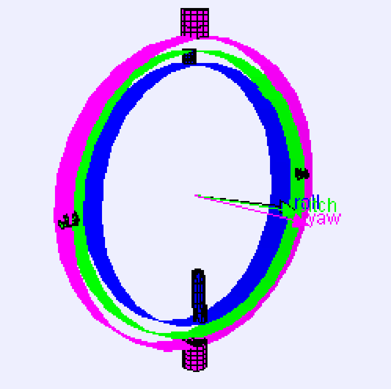


Вращение самолета углами Эйлера

Углы Эйлера II

Углы Эйлера в порядке y, x, z в виде матрицы

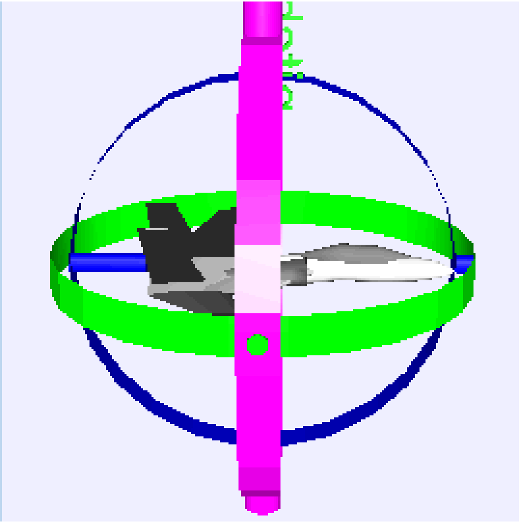
$$\boldsymbol{R} = \boldsymbol{R}_z(\gamma)\boldsymbol{R}_x(\beta)\boldsymbol{R}_y(\alpha) =$$
$$\begin{pmatrix} \mathsf{c}_\gamma\, \mathsf{c}_\alpha - \mathsf{s}_\gamma\, \mathsf{s}_\beta\, \mathsf{s}_\alpha & -\mathsf{c}_\beta\, \mathsf{s}_\gamma & \mathsf{c}_\gamma\, \mathsf{s}_\alpha + \mathsf{c}_\alpha\, \mathsf{s}_\gamma\, \mathsf{s}_\beta \\ \mathsf{c}_\alpha\, \mathsf{s}_\gamma + \mathsf{c}_\gamma\, \mathsf{s}_\beta\, \mathsf{s}_\alpha & \mathsf{c}_\gamma\, \mathsf{c}_\beta & \mathsf{s}_\gamma\, \mathsf{s}_\alpha - \mathsf{c}_\gamma\, \mathsf{c}_\alpha\, \mathsf{s}_\beta \\ -\mathsf{c}_\beta\, \mathsf{s}_\alpha & \mathsf{s}_\beta & \mathsf{c}_\beta\, \mathsf{c}_\alpha \end{pmatrix}$$



Вращение углами Эйлера

Углы Эйлера III

Складывание рамок (gimbal lock) — при $\beta = \pm 90^\circ$ изменение α и γ вращают объект вокруг одной оси (оси z объекта)



Углы Эйлера IV

- понятно человеку
- нет избыточности
- сложно применять
- сложно считать композицию, обратное вращение
- интерполяция дает специфичные* результаты
- плохо* подходит для оптимизационных задач

Угол + ось вращения I

Теорема вращения Эйлера: любое движение твёрдого тела в трёхмерном пространстве, имеющее неподвижную точку, является вращением тела вокруг некоторой оси.

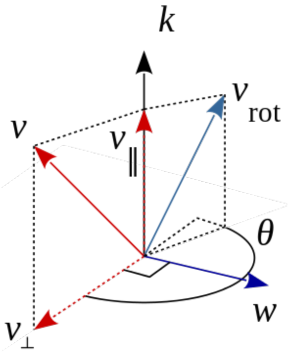
- $e = \begin{pmatrix} e_x & e_y & e_z \end{pmatrix}^T$ — ось вращения
- θ — угол

Угол + ось вращения II

Поворот вектора v на угол θ вокруг оси k (формула поворота Родрига)

$$v_{rot} = v\cos\theta + (k \times v)\sin\theta + k(k^\intercal v)(1 - \cos\theta)$$

Как из матрицы поворота $\boldsymbol{R}$ получить θ и k:
k — собственный вектор матрицы поворота,
$\boldsymbol{R}k = k$



Угол + ось вращения III

- не очень понятно человеку
- редко используется в чистом виде
- сложно применять
- легко считать композицию, обратное вращение
- интерполяция дает специфичные* результаты

Параметры Родрига

Пример параметров Родрига в OpenCV

- от оси вращения и угла к параметрам Родрига

$$r = \frac{e}{\|e\|}\theta$$

- от параметров Родрига к оси и углу

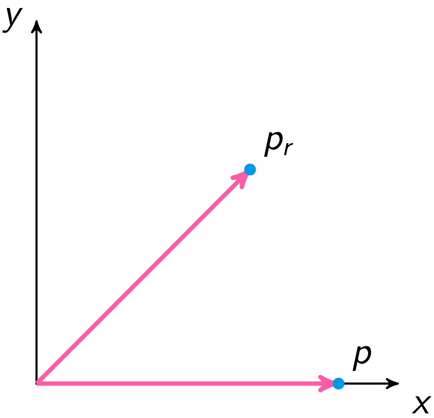
$$e = \frac{r}{\|r\|}$$
$$\theta = \|r\|$$

- по свойствам аналогичны параметризации угол + ось
- хорошо* подходит для оптимизационных задач

Интерполяция вращений в 2D I

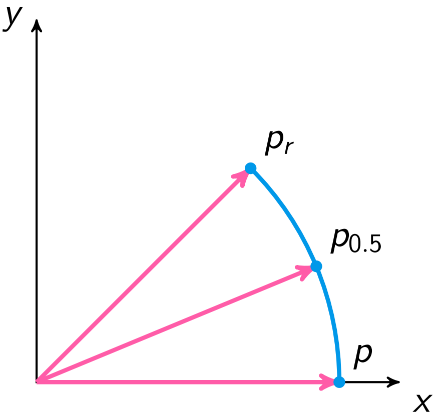
Интерполяция позиций объекта — получение промежуточных положений объекта между двумя позициями

Научимся разными способами интерполировать вращения в 2D и поймем, какие интерполяции и когда можно использовать

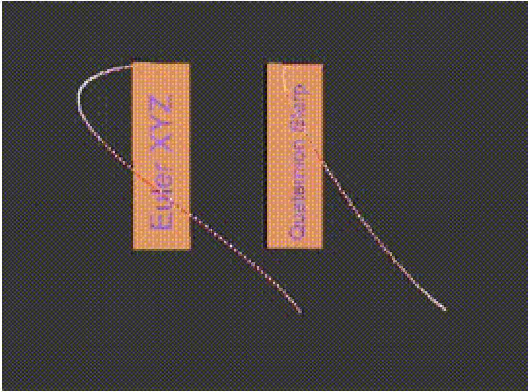


Интерполяция вращений в 2D II

- какую интерполяцию мы хотим?
- сферическая линейная интерполяция (Slerp)
- в 2D эквивалентна линейной интерполяции угла
- в 3D интерполяция углов Эйлера и угла в представлении угол + ось не дают нужный результат



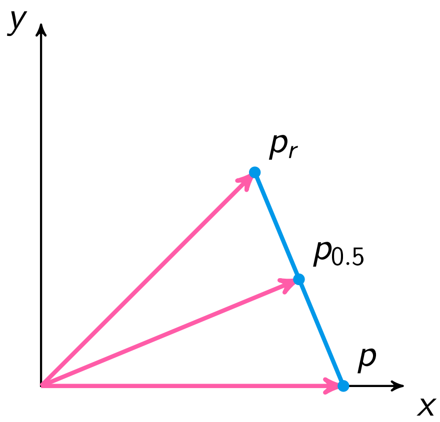
Интерполяция вращений в 2D III



Интерполяция углов Эйлера в сравнении с желаемой интерполяцией

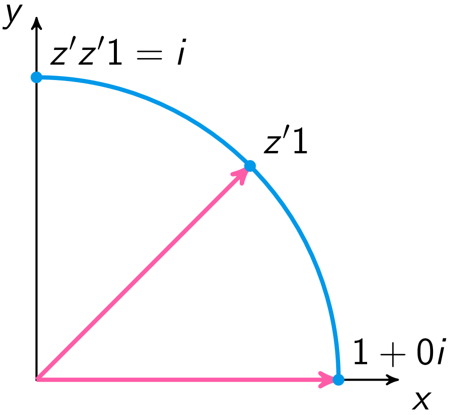
Интерполяция вращений в 2D IV

- интерполяция матриц поворота эквивалентна линейной интерполяции между p и p_r
- $p_t = ((1-t)\boldsymbol{R}_0 + t\boldsymbol{R}_1)p = (1-t)(\boldsymbol{R}_0 p) + t(\boldsymbol{R}_1 p)$
- $((1-t)\boldsymbol{R}_0 + t\boldsymbol{R}_1)$ — не матрица поворота



Комплексные числа и вращение в 2D I

- комплексное число $z = a + bi = re^{i\theta} = r(\cos\theta + i\sin\theta)$
- умножение на z — растяжение на r и поворот на θ
- $z_0 z_1 = r_0 r_1 e^{i(\theta_0+\theta_1)}$, при $r = 1$ комплексное число задает поворот
- $z' = \cos\frac{\pi}{4} + i\sin\frac{\pi}{4}$



Комплексные числа и вращение в 2D II

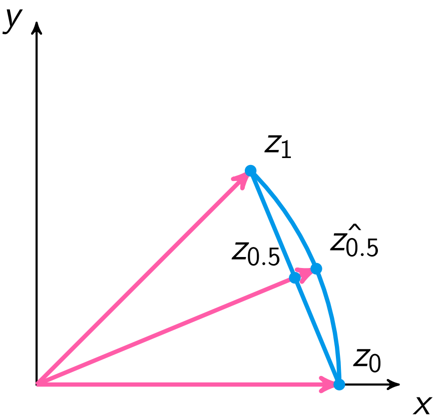
Операции с комплексными числами

- композиция поворотов: $z_2 = z_1 z_0$
- обратный поворот к $z = a + bi$: $z^{-1} = z^* = a - bi$
- применение вращения к p: умножение $p_x + ip_y$ на z
- все операции не требуют вычислений тригонометрических функций

Комплексные числа и вращение в 2D III

Интерполяция z_0, z_1

- линейная интерполяция: $z_t = (1-t)z_0 + tz_1$ — точка на линии (z_0, z_1)
- $\hat{z}_t = z_t / \|z_t\|$ — точка на дуге
- при равномерном движении по t угол $\hat{z}_t$ меняется неравномерно
- Slerp: $z'_t = \frac{\sin(1-t)\Omega}{\sin\Omega} z_0 + \frac{\sin t\Omega}{\sin\Omega} z_1$, где Ω — угол между z_0, z_1



Кватернионы I

- кватернионы $\mathbb{H}$ — четырехмерные гиперкомплексные числа

$$\begin{aligned}\mathbf{q} =& a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} = re^{\frac{\theta}{2}(e_x\mathbf{i}+e_y\mathbf{j}+e_z\mathbf{k})} \\ =& r(\cos\frac{\theta}{2} + \sin\frac{\theta}{2}(e_x\mathbf{i} + e_y\mathbf{j} + e_z\mathbf{k}))\end{aligned}$$

- $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$
- при $r = 1$, $\mathbf{q}$ задает недеформируемое движение четырехмерного пространства (изометрия)

Кватернионы II

Поворот в 3D с помощью кватернионов

- применение поворота к подпространству $b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ (формула поворота Родрига)

$$x_r\mathbf{i} + y_r\mathbf{j} + z_r\mathbf{k} = \mathbf{q} \cdot (x\mathbf{i} + y\mathbf{j} + z\mathbf{k}) \cdot \mathbf{q}^*$$

- $\mathbf{q} = \cos\dfrac{\theta}{2} + \sin\dfrac{\theta}{2}(e_x\mathbf{i} + e_y\mathbf{j} + e_z\mathbf{k})$ задает вращение 3D-точки на угол θ вокруг оси e
- $\mathbf{q}$ и $-\mathbf{q}$ задают одинаковый поворот

Кватернионы III

Операции с кватернионами

- композиция поворотов: $\mathbf{q}_2 = \mathbf{q}_1\mathbf{q}_0$
- обратный поворот к $\mathbf{q}$: $\mathbf{q}^{-1} = \mathbf{q}^* = a - bi - cj - dk$
- линейная интерполяция: $\mathbf{q}_t = (1-t)\mathbf{q}_0 + t\mathbf{q}_1$
- сферическая интерполяция: $\mathbf{q}_t = \frac{\sin(1-t)\Omega}{\sin\Omega}\mathbf{q}_0 + \frac{\sin t\Omega}{\sin\Omega}\mathbf{q}_1$
- все операции не требуют вычислений тригонометрических функций

Параметризация поворотов. Заключение

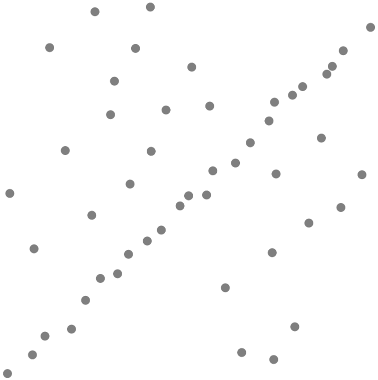
- существует множество параметризаций вращения
- матрицы поворота легко и удобно применять, используются для применения поворотов
- углы Эйлера понятны человеку, плохи для вычислений, используются как интерфейс для работы с вращениями
- параметры Родрига — хорошая параметризация для оптимизационных задач, используются в OpenCV
- кватернионы хороши для вычислений, используются для выполнения операций с поворотами

RANSAC I

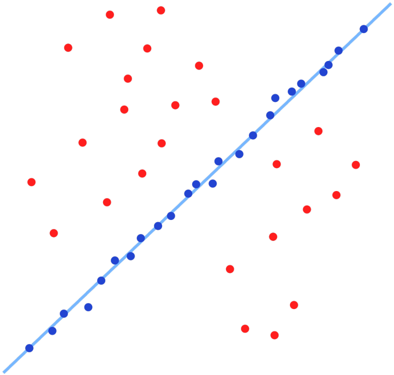
Random sample consensus (RANSAC) — алгоритм нахождения параметров модели (алгоритм нахождения максимального согласованного подмножества)

- Есть множество *измерений*. Есть параметризованная *модель*
- Необходимо найти параметры *модели*, такие что они удовлетворяют максимальному подмножеству *измерений*
- *инлаеры* — *измерения*, которые удовлетворяют найденным параметрам *модели*
- *выбросы* (*аутлаеры*) — *измерения*, которые не удовлетворяют найденным параметрам *модели*

RANSAC II



Измерения



Результат работы RANSAC

RANSAC III

Алгоритм RANSAC

- 1) возьмем случайное подмножество *измерений* — *выборку*
- 2) посчитаем на выборке параметры *модели* — *гипотезу*
- 3) на всем множестве *измерений* найдем количество *инлаеров*, удовлетворяющих *гипотезе*
- повторяем шаги 1-3 N раз. Будем искать *гипотезу*, соответствующую наибольшему количеству *инлаеров*

*как выбирать N?

Модификации RANSAC

- существуют различные модификации метода RANSAC
- PROSAC использует метрики качества данных для более эффективного перебора подмножеств
- MLESAC использует заданную оценку качества гипотезы вместо простого подсчета количества инлаеров