

ДОДАТКОВІ ЗАСТОСУВАННЯ

Розпізнавання руху на зображенні

Задача. На відео розпізнати рух деякого об'єкта.

Приклади застосування:

- а) Доповнена реальність: накладання зображення на об'єкт в русі.
- б) Розпізнавання рухомих об'єктів (людей, машин і т.п.)

Алгоритм знаходження оптичного потоку Лукаса-Канаде.

Беремо невелику область зображення. Нехай x, y – координати точки на зображенні, t – час, $I(x, y, t)$ – інтенсивність точки на чорно-білому зображенні.

Якщо деякий об'єкт (його центр) має траєкторію, що задається параметрично $x = u(t)$, $y = w(t)$, то маємо рівняння

$$I(u(t), w(t), t) = I_0,$$

де I_0 – інтенсивність об'єкта. Диференціюючи по t , отримаємо:

$$I'_x u'(t) + I'_y w'(t) + I'_t = 0.$$

Тут $\vec{v} = (u'(t), w'(t))$ – вектор швидкості руху об'єкта по зображенню, (I'_x, I'_y) – градієнт інтенсивності на зображенні (обчислюється для кожного пікселя наближено порівнянням з сусідніми).

Враховуючи помилки, мінімізуємо функцію

$$F(v_1, v_2) = \sum_{i=1}^n (I'_x(p_i)v_1 + I'_y(p_i)v_2 + I'_t(p_i))^2,$$

де p_i – пікселі зображення.

https://uk.wikipedia.org/wiki/Алгоритм_Лукаса_—_Канаде

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html

Розпізнавання площин

Припустимо, що роботом було знайдено низку особливих точок (фіч) і шляхом тріангуляції визначено їх положення в просторі. Отримуємо хмару з точок в просторі (сотні, тисячі точок). Задача - відновити форму об'єктів, на поверхні яких лежать точки.

Найпростіша задача - за частиною хмари побудувати площину (враховуючи помилки в глибинах). Якщо рівняння площини $x_3 = ax_1 + bx_2 + c$, то можна мінімізувати функцію

$$F(a, b, c) = \sum_{i=1}^n (ax_i + by_i + c - z_i)^2,$$

де (x_i, y_i, z_i) – точки хмари.

Є проблема знаходження підмножини точок, що лежать в одній площині. Існують підходи з ваговими функціями, що дозволяють відкинути невірні точки, наприклад

$$F(a, b, c) = \sum_{i=1}^n \arctg((ax_i + by_i + c - z_i)^2),$$

перетворення Хафа, а також алгоритми кластеризації.

<https://medium.com/swlh/how-to-find-the-least-squares-plane-from-a-cloud-of-point-using-excel-numbers-etc-92b66f852522>

<http://www.hinkali.com/Education/PointCloud.pdf>

https://en.wikipedia.org/wiki/Hough_transform

Положення робота-2

Робот орієнтується в просторі за допомогою сенсорів: акселерометра та гіроскопа. Знайти поточне положення.

Спрощена задача: поворот відсутній. Тоді орієнтуємося виключно по акселерометру.

Алгоритм: 1. Виділити гравітацію (наприклад, в момент спокою).

2. Двічі проінтегрувати прискорення і отримати рух:

$$v(t) = \int_0^t a(\tau) d\tau + v(0),$$

$$s(t) = \int_0^t v(\tau) d\tau + s(0).$$

Для чисельного інтегрування інтеграл заміняємо інтегральною сумою з приростом аргументу, що відповідає частоті сенсорів (порядку 1000 вимірів за секунду).

Код під андроїд для використання сенсорів:

https://developer.android.com/guide/topics/sensors/sensors_motion