

ОБРАБОТКА ИЗОБРАЖЕНИЙ ВИДЕОПОСЛЕДОВАТЕЛЬНОСТИ С ИСПОЛЬЗОВАНИЕМ ОПТИЧЕСКОГО ПОТОКА ДЛЯ ОТСЛЕЖИВАНИЯ ДВИЖЕНИЯ РОБОТА

Крицкий А. И.

Белорусский государственный университет, Минск, Беларусь,
e-mail: alexeykrytsky@gmail.com

Введение

Рассматривается проблема определения движения робота по видимому движению на изображениях, полученных с установленной на нем камеры. Это важная задача в компьютерном зрении, решение которой позволяет роботу самостоятельно ориентироваться в пространстве, контролировать правильность своего движения, сигнализировать о своих столкновениях с препятствиями.

Цель данной работы – подробно исследовать проблему определения движения робота по видимому с установленной на него камеры изображению и реализовать решение на практике, либо существующее, либо предложенное свое.

Постановка задачи

Пусть на подвижного робота установлена камера, имея жесткое соединение с его ходовой частью. Начальным положением O будем считать то положение центра камеры робота в мировой системе координат, в котором та находилась в момент времени первого кадра видеопоследовательности. Эту точку будем считать началом системы координат $OXYZ$ движения робота, где направление осей стандартно.

Задача: определить положение робота в системе координат $OXYZ$ в каждый из моментов времени отснятых кадров видеопоследовательности.

Допущения:

1) определим область точек неподвижной сцены как область, состоящую из точек, соответствующих определенному решению поставленной задачи (набору компонент векторов движения робота), которая занимает наибольшую площадь;

2) видимые сдвиги точек между двумя кадрами достаточно небольшие;

3) точка сохраняет свою интенсивность в течение малого промежутка времени.

Проведя рассуждения из [2], **геометрия** задачи сводится к следующему уравнению:

$$\mathbf{u}(x, y) = \frac{1}{z(x, y)} A(x, y) \mathbf{v} + B(x, y) \boldsymbol{\omega}, \quad (1)$$

где $A(x, y)$ и $B(x, y)$ – известные матрицы при известных (x, y) и f , $\mathbf{x} = (x, y) = \frac{f}{z}(X, Y)$ – проекция точки X на плоскость изображения, f – фокусное расстояние, $\mathbf{v} = (v_x, v_y, v_z)$ – переносная скорость точки O , $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ – ее угловая скорость, $\mathbf{u}(x, y) = (u(x, y), \vartheta(x, y))$ – оптический поток в точке (x, y) .

В работе будет рассматриваться частный случай, при котором камера совершает исключительно вращательное движение, тогда первое слагаемое в правой части уравнения зануляется.

Общая схема алгоритма решения задачи

Матричное уравнение (1) в данном случае можно рассматривать как задание отображения g , которое по известному решению (ω) каждой точке x изображения ставит в соответствие ее сдвиг $u(x)$ на следующем кадре видеопоследовательности так, как если бы эта точка принадлежала области неподвижной сцены.

$$g(\omega, x) = u. \quad (2)$$

Исходя из этого, предлагается следующая схема алгоритма решения поставленной задачи.

Пусть I – множество допустимых координат x точек изображения, а $I(x, t)$ – яркость пикселя с координатами x на t -ом кадре видеопоследовательности. Рассмотрим два кадра: t и $t+1$.

Шаг 1. Выбрать множество $W \subset I$ точек, для которых будет явно вычислен оптический поток.

Шаг 2. Вычислить оптический поток F для точек множества W .

Шаг 3. Сформировать семейство P двухэлементных подмножеств множества W : $P \subset \{P_i = \{p_1, p_2\} : p_1, p_2 \in W\}$.

Шаг 4. По каждому множеству (в рассматриваемом случае паре точек) P_i семейства P составить систему (1) и решить ее, в результате найдя соответствующее предполагаемое решение S_i .

Шаг 5. Составить множество точек $E \subset I$, которые будут участвовать в «голосовании» на следующем шаге.

Шаг 6. Завести список V , хранящий для каждого решения количество голосов за него. Организовать цикл по всем предполагаемым решениям S_i : для каждой точки p_j из E вычислить ожидаемый для решения S_i сдвиг u_j этой точки между рассматриваемыми кадрами видеопоследовательности, используя (1), и вычислить координаты сдвинутой точки $(p_j + u_j)$, а затем получить значение функции ошибки $\rho(p_j, p_j + u_j)$. Если для данного решения S_i значение $\rho(p_j, p_j + u_j) \leq \varepsilon$, то данное решение подходит для текущей точки p_j , и засчитать «голос» этой точки за это решение.

Шаг 7. Определить, какое из решений S_i выбрать в качестве верного решения поставленной задачи.

Реализация алгоритма и тестирование

Программа реализации алгоритма была написана на языке Python 3.7, использовались библиотеки `numpy`, `opencv`. Для вычисления оптического потока использовался метод Лукаса-Канаде, который хорошо описан в [1]. Его работа проверялась на изображениях из набора данных открытого ресурса «middlebury» [5].

Был подготовлен набор тестовых кадров, снятых с вращающейся на штативе камеры с известным углом, который составил приблизительно $\arctg(1/28) = 0.0356911$ радиан. Соответственно, ожидаемый вектор угловой скорости: $s = [0, 0.0356911, 0]$.

Эксперимент проводился для двух стратегий выбора множества S точек (рис. 1):

- 1) **grid**: выбираются точки на фиксированной сетке изображения;
- 2) **corners**: выбираются контурные точки. Первый подход показал лучший результат.

Табл. 1. Ошибка определения угла поворота вокруг оси OY, в радианах

Набор:	1: StaticThings	2: ForwardThings	3: BackwardThings
Grid	0,00192075	0,00005707	0,00081931
Corners	0,00217610	0,00092016	0,00129770

Сравним полученные результаты (таблица 1) с результатами алгоритмов, описанными в [4]. При тестировании на видеопоследовательности ошибка алгоритма Kanatani [3] колеблется обычно в пределах 0.02 – 0.46 радиан, Tomasi – в пределах 0.01 – 0.13, лишь в некоторых точках значительно сокращая ошибку ниже 0.01 радиана, а лучший достигнутый результат – 0,000348 радиана. Таким образом, даже простейшая реализация предложенного алгоритма дает хорошие результаты.

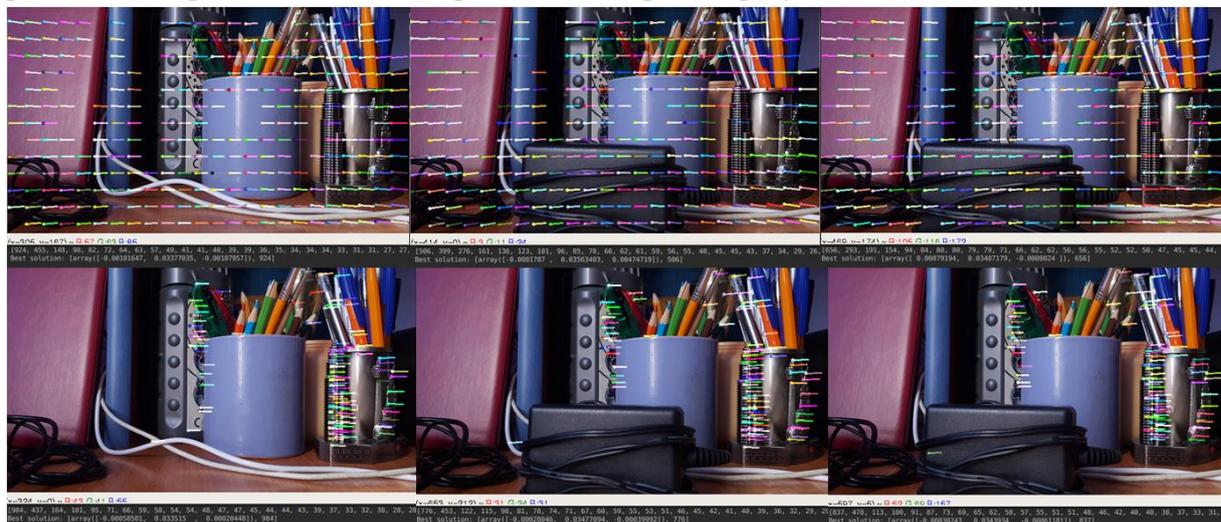


Рис. 1. Результаты тестирования реализации алгоритма на различных тестовых данных, по горизонтали: "StaticThings", "ForwardThings", "BackwardThings". Верхний ряд – стратегия «grid», нижний – «corners». Векторы показывают вычисленное по полученному решению ожидаемое видимое движение точек изображения в предположении, что они принадлежат области неподвижной сцены.

Заключение

Была достаточно глубоко рассмотрена проблема определения движения робота по видимому движению на изображениях, полученных с установленной на нем камеры. Как результат, была поставлена задача и предложена общая схема алгоритма ее решения для частного случая. Алгоритм был программно реализован и протестирован на подготовленных специально для рассматриваемого случая данных.

Литература

1. Bradski, G. Learning OpenCV : учеб. пос. / G. Bradski, A. Kaehler. – 1-е изд.-е. – США: O'Reilly Media, 2008. – 415 с. – С. 316–334.
2. A. R. Bruss, B.K.P. Horn.: Passive Navigation. Computer Vision, Graphics, and Image Processing 21, 3-20 (1983).
3. K. Kanatani.: 3-D Interpretation of Optical Flow by Renormalization. International Journal of Computer Vision, 267-282 (1993).
4. Пономарев, Е.С. Алгоритмы вычисления оптического потока в задаче определения собственного движения / Е.С. Пономарев, А.С. Григорьев // конференция «Информационные технологии и системы 2015». – 2015. – С. 464-470.
5. Optical Flow Datasets // Middlebury Optical Flow Page [Электронный ресурс]. – 2009. – Режим доступа: <http://vision.middlebury.edu/flow/data/>. – Дата доступа: 07.02.2020.