

ОПТИМИЗАЦИОННЫЙ ПОДХОД К РЕШЕНИЮ PnP-ЗАДАЧИ НА ОСНОВЕ ПАРАМЕТРИЗАЦИИ ВЕКТОРОМ РОДРИГА¹

Абраменков А. Н.²

(ФГБУН Институт проблем управления
им. В.А. Трапезникова РАН, Москва)

Рассматривается задача позиционирования подвижного объекта на основе изображений, получаемых с камеры. В ряде случаев данная задача может быть сведена к известной проблеме PnP (perspective-n-point), для решения которой в настоящей работе предлагается использовать оптимизационный алгоритм. В качестве функционала для минимизации рассматривается ошибка репроекции, использующая параметризацию матрицы поворота вектором Родрига. Предлагается подход с заменой переменных, для которого выводятся аналитические частные производные. Особенностью задачи позиционирования подвижного объекта можно считать то, что новое решение в большинстве случаев находится в окрестности предыдущего. Благодаря этому обходится основной недостаток оптимизационного подхода – необходимость вычислять начальное приближение. Также это позволяет обходить проблему с вектором Родрига, который неограниченно возрастает при приближении угла поворота к 180 градусам. Для решения оптимизационной задачи был использован численный алгоритм BFGS. Предложена эвристика для выбора начального приближения обратного гессиана и первого шага одномерного поиска, ускоряющая сходимость алгоритма. На синтетических данных проведено сравнение предложенного подхода с известными реализациями из библиотеки OpenCV. Эксперимент показал, что рассматриваемый подход обладает высокой скоростью вычисления, при этом не уступая по точности другим решениям.

Ключевые слова: визуальная одометрия, оптимизационный алгоритм, ошибка репроекции.

1. Введение

Необходимость определения положения подвижного объекта возникает во многих практических задачах. Часто для этого используют решения на основе глобальной навигационной

¹ Исследование выполнено за счет гранта Российского научного фонда № 23-19-00338, <https://rscf.ru/project/23-19-00338/>

² Александр Николаевич Абраменков, н.с. (aabramenkov@asmon.ru).

спутниковой системы (ГНСС). Однако не во всех случаях она бывает доступна. Например, при навигации подвижного объекта в помещении использование ГНСС становится невозможным. В этом случае возникает необходимость в альтернативных подходах. Среди них – визуальная одометрия, предполагающая для определения положения использование данных с видеокамеры. Более развитые SLAM-подходы (simultaneous localization and mapping) предполагают одновременно решать задачу позиционирования и построения карты.

Во многих случаях при рассмотрении подходов визуальной одометрии в том или ином виде возникает проблема PnP (perspective-n-point) [5, 20], которая заключается в определении положения камеры по известным положениям точек в трёхмерном пространстве и их проекциям на изображении. К данной проблеме также может быть сведена задача определения положения камеры в пространстве, размеченном специальными метками [13, 16]. В этом случае в качестве точек в трёхмерном пространстве, относительно которых решается задача, могут выступать, например, уголки меток. В работе [4] исследуются методы решения PnP-задачи для позиционирования относительно светодиодов с известными координатами.

Таким образом, PnP-проблема является важной подзадачей в задачах позиционирования. Существует большое количество подходов к решению данной проблемы, которые глобально можно разделить на итеративные и неитеративные.

Итеративные подходы в большинстве своём рассматривают PnP-задачу как оптимизационную и решают её с использованием методов Гаусса – Ньютона или Левенберга – Марквардта. Считается, что они обеспечивают лучшую точность, но при этом требуют больше вычислительных затрат [25, 19]. Главным недостатком этих методов является необходимость знать начальное приближение, которое потом итеративно уточняется. К этой группе можно отнести SQPnP [22], который использует схему последовательного квадратичного программирования, чтобы найти области локальных минимумов, с гарантией, что среди них будет глобальный.

Неитеративные подходы позволяют обеспечить нахождение решения, не используя начальное приближение. В минимально возможном случае трёх точек (P3P) при отсутствии шума задача может иметь аналитическое решение, описывающее до четырёх геометрически реализуемых конфигураций [9]. Задача P3P может являться составной частью других решений. Например, используя схему RANSAC [7], она может быть обобщена до большего количества точек. Более общими и наиболее известными подходами являются DLS [11] и EPnP [14].

Неитеративные подходы могут быть объединены с оптимизационными схемами. Например, авторы EPnP рекомендуют использовать результат их алгоритма как начальное приближение для последующей оптимизации [14]. Ряд подходов в качестве одного из шагов могут использовать метод Гаусса – Ньютона для уточнения результата. К ним можно отнести CPnP [24], Hidden PnP [19] и ACEPnP [21], который является развитием EPnP. В работе [26] для дальнейшего уточнения решения используют алгоритм L-BFGS.

Часть подходов акцентируют внимание на наличии шумов в измеряемых данных, например, GMLPnP [25] и EPnPU [23]. В работе [25] предлагается метод, который использует итеративную процедуру для одновременного нахождения позиции и оценки неопределённости.

Некоторые подходы рассматривают решение частных задач для заданного количества точек. Например, в [3] рассматривается задача P2P для определения угловых и пространственных координат по двум реперным точкам. В работе [1] рассматривается случай 4 точек (P4P). В [2] решение задачи P4P используется для определения позиции относительно реперных точек.

Повышение вычислительной эффективности методов решения PnP задачи может быть важным при использовании на маломощных бортовых вычислительных системах. Для этого, например, в работе [15] предлагается реализация решения на ПЛИС.

Особенностью определения положения по данным с видеокамеры является то, что положение подвижного объекта не мо-

жет меняться моментально. Это предположение можно использовать, чтобы обойти основной недостаток оптимизационного подхода, беря в качестве начального приближения результат, полученный в предыдущий момент времени.

В данной работе для решения проблемы PnP предлагается использовать оптимизационный алгоритм BFGS (Broyden-Fletcher-Goldfarb-Shanno). Далее будет показано, что такой подход при определённых условиях способен демонстрировать высокую производительность, сравнимую с лучшими решениями, не уступая им по точности.

2. Предлагаемый подход

2.1. ИСПОЛЬЗУЕМЫЕ ОБОЗНАЧЕНИЯ

Все скалярные величины будем обозначать строчными буквами, а векторы и матрицы – заглавными.

Для описания построения плоского изображения наблюдаемого реального трехмерного пространства будем использовать модель камеры-обскуры [10]. На рис. 1 приведена геометрическая интерпретация данной модели.

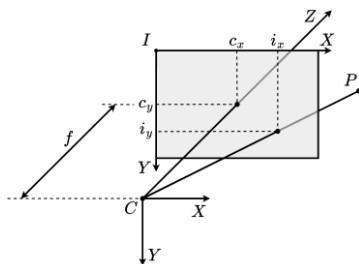


Рис. 1. Геометрическая интерпретация математической модели камеры-обскуры

Точка $P = [p_x \ p_y \ p_z]^T$ из трехмерного пространства с координатами, заданными в системе координат (СК) камеры $CXYZ$, проецируется в точку на плоском изображении, имеющую координаты $[i_x \ i_y]^T$ в СК плоскости изображения (экрана) IXY . Координаты проекции описываются следующим соотношением:

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \frac{f}{p_z} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}.$$

Здесь $[i_x \ i_y]^T$ – координаты проекции точки, отсчитанные от верхнего левого угла изображения; $[c_x \ c_y]^T$ – координаты центра изображения; f – расстояние от центра изображения до отверстия камеры-обскуры.

Поведение реальной оптической системы не может быть точно описано элементарной моделью камеры-обскуры, но при разработке методов решения PNP-задачи этим обстоятельством пренебрегают [10].

В дальнейшем для описания проекций точек трёхмерного пространства на плоскость изображения будем использовать нормированные безразмерные координаты:

$$N = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \frac{1}{f} \begin{bmatrix} i_x - c_x \\ i_y - c_y \end{bmatrix}.$$

Здесь $[n_x \ n_y]^T$ – нормированные координаты проекции точки. Таким образом из дальнейших выражений исключаются параметры математической модели камеры (f и $[c_x \ c_y]^T$).

Введём оператор проецирования H , который отображает на изображение точку $P = [p_x \ p_y \ p_z]^T$ из трёхмерного пространства в СК камеры:

$$N = H(P) = \frac{1}{p_z} \begin{bmatrix} p_x \\ p_y \end{bmatrix}.$$

Для описания положения и ориентации объекта будем использовать пару M , состоящую из вектора линейного переноса T и матрицы поворота R :

$$M = \{T, R\}.$$

Преобразование точки P из трёхмерного пространства описывается следующим выражением:

$$MP = RP + T.$$

Ниже приведены основные соотношения для M :

$$M_1 M_2 = \{R_1 T_2 + T_1, R_1 R_2\},$$

$$M^{-1} = \{-R^{-1}T, R^{-1}\}.$$

2.2. ПОСТАНОВКА ЗАДАЧИ

Пусть в выбранной СК известно положение m точек. Также известны их нормированные координаты на изображении. Необходимо определить положение камеры M_c в выбранной СК мира. Данная задача может быть сформулирована как оптимизационная, если ввести функционал ошибки e_r следующего вида:

$$(1) \quad e_r = \frac{1}{m} \sum_k \|N_k - H(M_p P_k)\|^2.$$

Здесь N_k – измеренная проекция k -ой точки; P_k – положение k -й точки в выбранной СК. Решением является неизвестный параметр M_p , при котором функционал достигает минимума. Требуемое положение камеры определяется выражением

$$M_c = M_p^{-1}.$$

Выражение (1) можно переписать таким образом, чтобы решение искалось как изменение начального приближения. Для этого введём обозначение:

$$M_p = M'_p M_0.$$

Здесь M_0 – начальное приближение; M'_p – изменение решения. Теперь ошибка e_r будет иметь вид

$$(2) \quad e_r = \frac{1}{m} \sum_k \|N_k - H(M'_p P'_k)\|^2.$$

Здесь $P'_k = M_0 P_k$ – положение точки в трёхмерном пространстве относительно начального приближения.

Если использовать предположение о небольшом перемещении подвижного объекта и в качестве начального приближения брать результат с предыдущего момента времени, то минимум выражения (2) будет достигаться при значении параметров близких к нулю.

Самое первое начальное приближение может быть получено любым доступным алгоритмом PnP, например, SQPnP.

2.3. ВЕКТОР РОДРИГА

Матрица поворота содержит девять переменных, поэтому её обычно параметризуют представлением вращения меньшей

размерности. Среди них наиболее известны углы Эйлера и кватернион. В данной работе для описания вращения будет использоваться вектор Родрига [6]:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \operatorname{tg}\left(\frac{a}{2}\right) \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}.$$

Здесь $[r_x \ r_y \ r_z]^T$ – элементы вектора Родрига; $[u_x \ u_y \ u_z]^T$ – единичный вектор, вокруг которого осуществляется поворот на угол a . Благодаря такому представлению, матрица поворота будет иметь вид

$$(3) \quad R = \frac{1}{s} \begin{bmatrix} 1 + r_x^2 - r_y^2 - r_z^2 & 2(r_x r_y - r_z) & 2(r_x r_z + r_y) \\ 2(r_x r_y + r_z) & 1 - r_x^2 + r_y^2 - r_z^2 & 2(r_y r_z - r_x) \\ 2(r_x r_z - r_y) & 2(r_y r_z + r_x) & 1 - r_x^2 - r_y^2 + r_z^2 \end{bmatrix}.$$

Здесь $s = 1 + r_x^2 + r_y^2 + r_z^2$. Данное выражение может быть получено, например, с использованием формулы для матрицы поворота через единичный вектор и угол [6].

Выражение (3) является вычислительно эффективным представлением, так как требует всего 3 параметра и не использует тригонометрических функций. Единственный недостаток заключается в том, что при приближении угла a к 180 градусам элементы вектора Родрига неограниченно возрастают. Данный недостаток легко обходится, если искать решение относительно начального приближения, как было предложено ранее.

Раскроем функцию проекции H из выражения (1) и подставим в него матрицу поворота (3), получим следующие выражения:

$$\begin{aligned} x_k &= (1 + r_x^2 - r_y^2 - r_z^2)p_{xk} + 2(r_x r_y - r_z)p_{yk} + 2(r_x r_z + r_y)p_{zk} + st_x, \\ (4) \quad y_k &= 2(r_x r_y + r_z)p_{xk} + (1 - r_x^2 + r_y^2 - r_z^2)p_{yk} + 2(r_y r_z - r_x)p_{zk} + st_y, \\ z_k &= 2(r_x r_z - r_y)p_{xk} + 2(r_y r_z + r_x)p_{yk} + (1 - r_x^2 - r_y^2 + r_z^2)p_{zk} + st_z. \end{aligned}$$

Здесь: $P_k = [p_{xk} \ p_{yk} \ p_{zk}]^T$; $T = [t_x \ t_y \ t_z]^T$ – неизвестный вектор линейного перемещения. Теперь ошибка k -й точки по каждой координате будет иметь вид

$$e_{xk} = n_{xk} - x_k / z_k; \quad e_{yk} = n_{yk} - y_k / z_k.$$

Ниже приведена итоговая запись ошибки репроекции с учётом введённых обозначений:

$$(5) \quad e_r = \frac{1}{m} \sum_k^m (e_{xk}^2 + e_{yk}^2).$$

Можно заметить, что делитель s из (3) в выражении (4) стал множителем к неизвестному вектору линейного перемещения. Сделаем в (4) следующую замену переменных:

$$[t'_x \ t'_y \ t'_z]^T = s[t_x \ t_y \ t_z]^T.$$

В данной работе новые неизвестные переменные со штрихом предлагается считать независимыми от элементов вектора Родрига, что сделает последующий расчёт аналитических частных производных немного проще. Далее на экспериментальных данных будет показано, что такое допущение не влияет на точность находимого решения.

2.4. АНАЛИТИЧЕСКИЕ ЧАСТНЫЕ ПРОИЗВОДНЫЕ

Продифференцируем (5) по неизвестным переменным. Итоговый результат с учётом группировки для минимизации количества вычислений будет иметь вид:

$$(6) \quad \begin{aligned} \partial e_r / \partial t'_x &= -(2/m) \sum_k^m e'_{xk}, \\ \partial e_r / \partial t'_y &= -(2/m) \sum_k^m e'_{yk}, \\ \partial e_r / \partial t'_z &= (2/m) \sum_k^m e'_{xyk}, \\ \partial e_r / \partial r_x &= -(4/m) \sum_k^m (e'_{xk} s_{0k} - e'_{yk} s_{3k} - e'_{xyk} s_{2k}), \\ \partial e_r / \partial r_y &= -(4/m) \sum_k^m (e'_{xk} s_{3k} + e'_{yk} s_{0k} + e'_{xyk} s_{1k}), \\ \partial e_r / \partial r_z &= (4/m) \sum_k^m (e'_{xk} s_{2k} - e'_{yk} s_{1k} + e'_{xyk} s_{0k}). \end{aligned}$$

В выражении (6) используются следующие обозначения:

$$e'_{xk} = e_{xk} / z_k; \quad e'_{yk} = e_{yk} / z_k; \quad e'_{xyk} = (e'_{xk} + e'_{yk}) / z_k,$$

$$s_{0k} = p_{xk} r_x + p_{yk} r_y + p_{zk} r_z,$$

$$s_{1k} = p_{xk} - p_{yk} r_z + p_{zk} r_y,$$

$$s_{2k} = p_{yk} + p_{xk} r_z - p_{zk} r_x,$$

$$s_{3k} = p_{zk} - p_{xk} r_y + p_{yk} r_x.$$

2.5. АЛГОРИТМ BFGS

В данной работе для решения оптимизационной задачи предлагается использовать BFGS алгоритм [17]. Он относится к классу квазиньютоновских методов, не требующих вычисления матрицы Гессе в явном виде. Среди особенностей BFGS можно выделить, что в процессе выполнения алгоритма вычисляется оценка сразу для обратного гессиана. Это позволяет обойтись без процедуры обращения матрицы. Несмотря на то, что рассматриваемая задача имеет небольшую размерность (6 неизвестных), эта особенность может дать дополнительный прирост производительности.

Алгоритм BFGS может быть описан следующей последовательностью шагов. Сначала выбирается начальное приближение для решения X_0 и обратной матрицы Гессе B_0^{-1} , вычисляются вектор частных производных ∂F_0 функции в точке X_0 . Устанавливается номер итерации $k = 0$.

В качестве неизвестных параметров X в данном случае выступают значения линейного переноса и параметры матрицы поворота. Так как решение ищется относительно начального приближения M_0 , то X_0 берётся равной нулю. В качестве функции f используется выражение (5), а её частные производные ∂F приведены в выражении (6).

Первый шаг заключается в определении направления поиска решения D_k , для чего используется оценка обратного гессиана:

$$D_k = -B_k^{-1} \partial F_k.$$

На втором шаге, используя процедуру одномерного поиска, выбирается величина шага a_k , удовлетворяющего условию Вольфе [17]. Для этого используется аппроксимация функции кубическим полиномом и в качестве нового пробного шага берётся значение, при котором полином достигает минимума [17]. Поиск продолжается до тех пор, пока не будет выполнено условие Вольфе.

Вычисляется новое решение:

$$X_{k+1} = X_k + a_k D_k.$$

На третьем шаге, используя значения частных производных в новой точке ∂F_{k+1} , обновляется оценка обратного гессиана B_k^{-1} . Для этого применяется формула [8], которая достаточно эффективна в вычислительном плане, так как может быть реализована без умножений матрицы на матрицу:

$$B_{k+1}^{-1} = B_k^{-1} + \frac{S_k S_k^T}{S_k^T Y_k} \left(1 + \frac{Y_k^T B_k^{-1} Y_k}{S_k^T Y_k} \right) - \frac{S_k Y_k^T B_k^{-1} + B_k^{-1} Y_k S_k^T}{S_k^T Y_k}.$$

Здесь $S_k = X_{k+1} - X_k$; $Y_k = \partial F_{k+1} - \partial F_k$.

После обновления обратного гессиана алгоритм повторяется с первого шага, изменяя номер текущей итерации на $k = k + 1$. Критерием остановки является достижение нормы ∂F_k некоторого порога или ограничение на общее количество итераций.

Важным аспектом для эффективной работы алгоритма является выбор хорошего начального приближения для B_0^{-1} . Часто для этого используют единичную матрицу. В данной работе предлагаются два подхода к ускорению алгоритма.

Первый подход заключается в эмпирическом подборе B_0^{-1} , который позволит в большинстве случаев ускорить сходимость алгоритма применительно к рассматриваемой задаче. Для этого предлагается использовать диагональную матрицу с разными значениями коэффициентов:

$$(7) \quad B_0^{-1} = \text{diag}\{b_1, b_1, b_1, b_2, b_2, b_2\}.$$

Далее на экспериментальных данных, будет показано, что хорошим выбором может являться $b_1 = 20$ и $b_2 = 0,5$.

Второй подход заключается в модификации алгоритма одномерного поиска. Обычно для первого пробного шага используют $a_k = 1$. В данной работе предлагается следующая эвристическая зависимость:

$$(8) \quad a_k = \min \left(2 \left| \frac{f_k}{\partial F_k P_k} \right|, 1 \right).$$

Здесь f_k – значение функции в точке X_k .

На экспериментальных данных будет показано, что такая модификация может давать небольшой прирост производительности.

В работе применялась самостоятельно написанная на языке программирования C++ реализация BFGS алгоритма [12], позволяющая использовать предложенные подходы.

3. Эксперимент на синтетических данных

3.1. ОПИСАНИЕ ЭКСПЕРИМЕНТА

Для сравнения предложенных подходов использовались синтетически полученные данные. Ниже приведён алгоритм их генерации.

Метрические параметры математической модели камеры, в соответствии с требованиями распространенных библиотек и пакетов программ машинного зрения [18], задавались в пикселях и составили значения: размер изображения 1280×720 пикселей, $f = 600$ пикселей, что соответствует углу обзора примерно в 90 градусов. Изменение этих параметров в определённых пределах качественно не влияло на результат.

На изображении случайным образом создавалось требуемое количество проекций точек. Для каждой точки случайным образом из диапазона 0,5–10 м определялось расстояние до камеры. Полученные точки случайным образом перемещались и поворачивались в пространстве. Координаты вектора линейного перемещения выбирались из диапазона от $-0,5$ до $0,5$ м. Поворот осуществлялся вокруг случайной оси на угол из диапазона от $-0,5$ до $0,5$ рад.

Для каждой проекции точки добавлялся шум, представляющий из себя изменение координаты на случайную величину из диапазона от $-\delta$ до δ пикселей. Приведённые далее результаты были получены при значении $\delta = 1$. Увеличение этого параметра качественно не влияло на результат и приводило к пропорциональному увеличению величины ошибок. Особым случаем является отсутствие шумов, чего обычно не бывает в реальных данных, и поэтому далее не рассматривалось.

Для ошибки по расстоянию e_p использовалось следующее выражение:

$$e_p = \|T_r - T_m\|.$$

Здесь T_r – вектор истинного значения перемещения; T_m – вектор рассчитанного значения перемещения.

Метрика ошибки поворота e_a представляет собой угол кратчайшего поворота между истинной и рассчитанной ориентацией. Для её вычисления использовалось следующее выражение:

$$Q_d = [q_w, q_x, q_y, q_z] = Q_r Q_m^{-1},$$

$$e_a = 2 \arccos (q_w).$$

Здесь: Q_r – кватернион истинной ориентации; Q_m – кватернион рассчитанной ориентации; Q_d – кватернион разности двух ориентаций.

Каждая итерация теста заключалась в генерации 10^6 наборов данных, на которых решалась PnP-задача. Для каждого набора вычислялись значения ошибок и засекалось время выполнения алгоритма. Для сравнения использовались медианные значения.

Все вычисления проводились на компьютере с процессором Intel Core i5-9300H под управлением операционной системы Ubuntu-18.

3.2. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

На рис. 2 приведены значения ошибок в зависимости от количества точек при использовании предложенной замены переменных (b) и без (a). В этих случаях частные производные вычислялись численно. В дополнение на графиках также представлен вариант с аналитическими частными производными (c). Результаты получены без использования предложенной модификации первого шага одномерного поиска и с единичной матрицей в качестве начального приближения обратного гессiana. Видно, что с точки зрения точности все варианты демонстрируют практически идентичные показатели. Из этого можно сделать вывод, что подход с заменой переменных не влияет на точность решения, а аналитические частные производные выведены корректно.

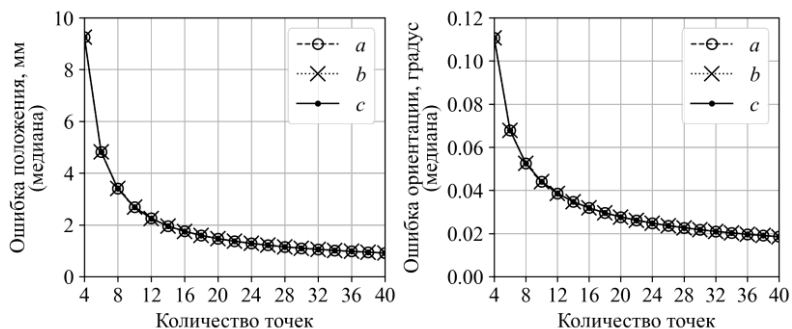


Рис. 2. Значения ошибок для случая без замены переменных (а), с заменой (б) и с аналитическими производными (с)

На рис. 3 для рассмотренных вариантов приведено время расчёта. Вариант с заменой переменных оказался незначительно быстрее, чем без замены. Использование аналитических производных позволило практически в два раза ускорить алгоритм.

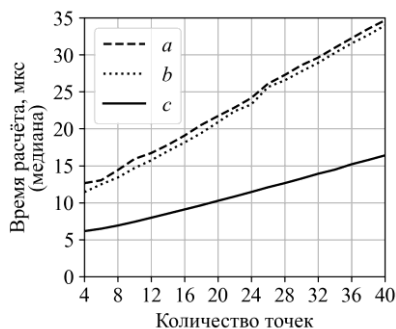


Рис. 3. Время расчёта для случая без замены переменных (а), с заменой (б) и с аналитическими производными (с)

Далее рассмотрим влияние выбора начального приближения для обратной матрицы Гессе и предложенного подхода для выбора первого шага одномерного поиска. Эксперимент проводился на рассмотренном варианте с заменой переменных и аналитическими частными производными (б). На рис. 4 приведены результаты медианного времени расчёта для нескольких вари-

антов. Графики *a* и *b* соответствуют выбору в качестве начального приближения для B_0^{-1} единичной матрицы, а графики *c* и *d* – предложенному выбору согласно выражению (7) с эмпирически подобранными коэффициентами $b_1 = 20$ и $b_2 = 0,5$. Видно, что такой подход обеспечивает повышение вычислительной эффективности алгоритма. Для вариантов *b* и *d* была использована предложенная зависимость (8) для модификации первого шага одномерного поиска, а варианты *a* и *c* соответствуют стандартному выбору. Здесь также видно, что предложенная модификация первого шага одномерного поиска более эффективна в вычислительном плане. Совместное использование обоих предложенных подходов (*d*) демонстрируют наилучшее время расчёта.

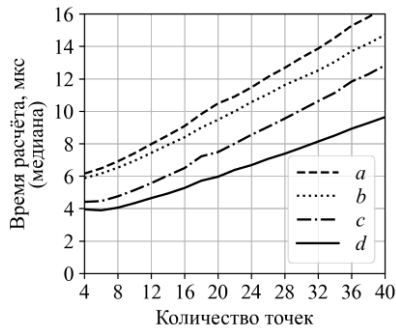


Рис. 4. Время расчёта для стандартного выбора начального приближения обратного гесссиана (*a*, *b*) и предложенного (*c*, *d*), с модификацией первого шага одномерного поиска (*b*, *d*) и без (*a*, *c*)

Показатели точности для всех вариантов в результате эксперимента были практически одинаковыми, повторяя результат на рис. 2.

3.3. СРАВНЕНИЕ С СУЩЕСТВУЮЩИМИ ПОДХОДАМИ

Для сравнения с существующими подходами решения проблемы PnP были выбраны несколько алгоритмов из библиотеки компьютерного зрения OpenCV [18]. Она является наиболее из-

вестной библиотекой с открытым исходным кодом и активно развивается огромным сообществом. Написана на языке программирования C++ и повсеместно используется во многих программных решениях. Это позволяет предполагать, что реализованные в ней алгоритмы хорошо оптимизированы. Ниже кратко рассмотрим выбранные алгоритмы и то, как они будут обозначаться.

SQPnP является одним из самых быстрых алгоритмов по приведённым в [25] результатам. При этом по точности он уступал только GMLPnP, который оказался заметно медленнее.

EPnP является самым известным неитеративным алгоритмом. Его часто используют для сравнения в различных публикациях [19, 22, 24, 25].

LM – итеративный подход, использующий оптимизационный алгоритм Левенберга – Марквардта.

BFGS – предложенное решение, использующее аналитические частные производные и рассмотренные подходы к повышению скорости.

На рис. 5 приведены результаты сравнения точности рассматриваемых решений. Предложенный подход демонстрирует лучшую точность, аналогичную LM.

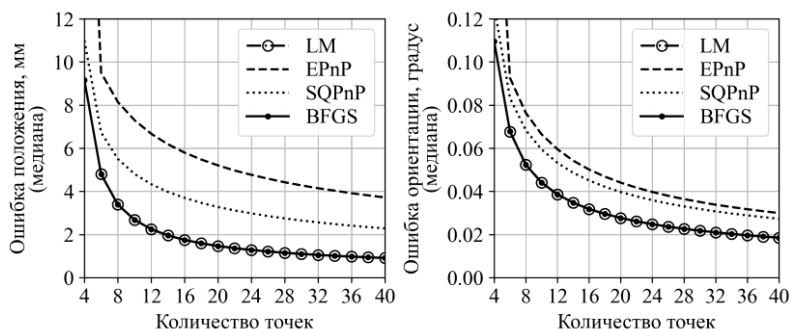


Рис. 5. Значения ошибок для сравниваемых решений

На рис. 6 приведены результаты времени расчёта где предложенное решение и SQPnP демонстрируют заметно более высокую производительность.

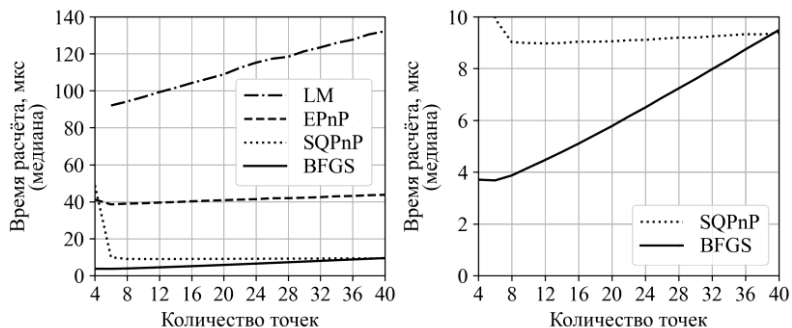


Рис. 6. Время расчёта для сравниваемых решений при разном масштабе по оси ординат

Видно, что производительность предложенного решения сильнее зависит от количества точек, чем SQPnP. По результатам данного эксперимента на 40 точках предложенное решение начинает уступать по скорости SQPnP, при этом оставаясь всё ещё более точным.

Стоит отметить особый случай четырёх точек, на которых LM не работает, а EPnP демонстрирует наиболее плохие результаты по точности. Производительность SQPnP на четырёх точках заметно ухудшается. При этом предложенное решение не демонстрирует никаких особых изменений в поведении.

4. Заключение

В работе было представлено оптимизационное решение PnP-задачи с использованием BFGS алгоритма. Рассмотрены различные способы повышения вычислительной эффективности. Среди них – вариант параметризации поворота через вектор Родрига и вывод аналитических частных производных. Было предложено два подхода к модификации BFGS алгоритма, которые в рамках рассматриваемой задачи привели к повышению скорости вычисления. Благодаря этому предложенное решение демонстрирует высокую производительность при сохранении точности, что может быть полезно при решении задач позицио-

нирования на маломощных бортовых системах подвижных объектов.

Стоит отметить, что если в процессе вычисления на каком-либо шаге будет получено некорректное решение, то дальнейшее его использование в качестве начального приближения может привести к непредсказуемым результатам. Анализ данной ситуации требует дальнейшего исследования.

Литература

1. БЕКЛЕМИШЕВ Н.Д. *Прямое нахождение оценки положения камеры центральной проекции по четырем опорным точкам* // Математическое моделирование. – 2020. – Т. 32, №10. – С. 91–104.
2. КУДИНОВ И.А., ПАВЛОВ О.В., ХОЛОПОВ И.С. *Реализация алгоритма определения пространственных координат и угловой ориентации объекта по реперным точкам, использующего информацию от одной камеры* // Компьютерная оптика. – 2015. – Т. 39, №3. – С. 413–419.
3. ХОЛОПОВ И.С., КАЛИНКИН А.И. *Оценка погрешности определения угловых координат объекта с двумя реперными излучателями* // Вестник Рязанского государственного радиотехнического университета. – 2019. – №69. – С. 52–59.
4. APARICIO-ESTEVE E., URENA J., HERNANDEZ A. et al. *Using Perspective-n-Point Algorithms for a Local Positioning System Based on LEDs and a QADA Receiver* // Sensors. – 2021. – Vol. 21, No. 19. – 6537.
5. CUI J., MIN C., FENG D. *Research on pose estimation for stereo vision measurement system by an improved method: uncertainty weighted stereopsis pose solution method based on projection vector* // Optics Express. – 2020. – Vol. 28, No. 4. – P. 5470–5491.
6. DAI J.S. *Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections* // Mechanism and Machine Theory. – 2015. – Vol. 92. – P. 144–152.
7. FISCHLER M.A., BOLLES R.C. *Random sample consensus: a paradigm for model fitting with applications to image analysis*

- and automated cartography* // Communications of the ACM. – 1981. – Vol. 24, No. 6. – P. 381–395.
8. FLETCHER R. *Practical Methods of Optimization*. – Wiley, 2000. – 436 p.
 9. GAO X.S., HOU X.R., TANG J. et al. *Complete solution classification for the perspective-three-point problem* // IEEE Trans. on Pattern Analysis and Machine Intelligence. – 2003. – Vol. 25, No. 8. – P. 930–943.
 10. HARTLEY R., ZISSERMAN A. *Multiple View Geometry in Computer Vision*. – New York: Cambridge University Press, 2003.
 11. HESCH J.A., ROUMELIOTIS S.I. *A Direct Least-Squares (DLS) method for PnP* // 2011 Int. Conf. on Computer Vision. – 2011.
 12. *Implementation of the BFGS algorithm in C++* [Электронный ресурс]. – URL: <https://github.com/IOdissey/bfgs> (дата обращения: 27.03.2025).
 13. KAWECKI A., DĄBROWSKI P., JANUSZKO S. et al. *AR Tags Based Absolute Positioning System* // 8th Int. Conf. on Automation, Robotics and Applications (ICARA). – 2022. – P. 62–67.
 14. LEPETIT V., MORENO-NOGUER F., FUA P. *EPnP: An Accurate $O(n)$ Solution to the PnP Problem* // Int. Journal of Computer Vision. – 2008. – Vol. 81. – P. 155–166.
 15. LV H., WU Q. *An Energy-Efficient Field-Programmable Gate Array (FPGA) Implementation of a Real-Time Perspective-n-Point Solver* // Electronics. – 2024. – Vol. 13, No. 19. – 3815.
 16. *Map-based navigation with ArUco markers* [Электронный ресурс]. – URL: https://clover.coex.tech/en/aruco_map.html (дата обращения: 24.03.2025).
 17. NOCEDAL J., WRIGHT S.J. *Numerical Optimization*. – New York: Springer New York, 2006. – 664 p.
 18. *OpenCV – Open Computer Vision Library* [Электронный ресурс]. – URL: <https://opencv.org/> (дата обращения: 12.08.2025).
 19. QIAO R., XU G., WANG P. et al. *An Accurate, Efficient, and Stable Perspective-n-Point Algorithm in 3D Space* // Applied Sciences. – 2023. – Vol. 13, No. 2. – 1111.

20. SCARAMUZZA D., FRAUNDORFER F. *Visual Odometry [Tutorial]* // IEEE Robotics & Automation Magazine. – 2011. – Vol. 18, No. 4. – P. 80–92.
21. SUN Q., ZHANG T., ZHANG G. et al. *Efficient Solution to PnP Problem Based on Vision Geometry* // IEEE Robotics and Automation Letters. – 2024. – Vol. 9, No. 4. – P. 3100–3107.
22. TERZAKIS G., LOURAKIS M. *A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem* // European Conf. on Computer Vision. – 2020. – P. 478–494.
23. VAKHITOV A., FERRAZ L., AGUDO A et al. *Uncertainty-aware camera pose estimation from points and lines* // Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). – 2021. – P. 4659–4668.
24. ZENG G., CHEN S., MU B. et al. *CPnP: Consistent Pose Estimator for Perspective-n-Point Problem with Bias Elimination* // IEEE Int. Conf. on Robotics and Automation (ICRA). – 2023. – P. 1940–1946.
25. ZHAN T., XU C., ZHANG C. et al. *Generalized Maximum Likelihood Estimation for Perspective-n-Point Problem* // IEEE Robotics and Automation Letters. – 2025. – Vol. 10, No. 2. – P. 1752–1759.
26. ZHUANG S., ZHAO Z., CAO L. et al. *A Robust and Fast Method to the Perspective-n-Point Problem for Camera Pose Estimation* // IEEE Sensors Journal. – 2023. – Vol. 23, No. 11. – P. 11892–11906.

OPTIMIZATION APPROACH TO SOLVING PNP PROBLEM BASED ON PARAMETERIZATION BY RODRIGUES VECTOR

Alexander Abramnikov, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Researcher (aabramnikov@asmon.ru).

Abstract: In many cases, the task of positioning a moving object using camera images can be reduced to well-known PnP (perspective-n-point) problem. For solving it, optimization algorithm is considered. The reprojection error using the parameterization of the rotation matrix by the Rodrigues vector is considered. Analytical partial derivatives were obtained for it. A feature of the task of positioning a moving object is that the new solution in most cases is in the neighborhood of the previous

one. The main disadvantage of the optimization approach is the need to calculate the initial solution. It can be eliminated using the solution from the previous iteration. This also helps to eliminate the problem with the Rodrigues vector, which increases indefinitely as the rotation angle reaches 180 degrees. The BFGS numerical algorithm was used to solve the optimization problem. A heuristic for selecting the initial approximation of the inverse Hessian and the first step of the linear search is proposed. This made it possible to speed up the algorithm. The proposed approach is compared with known implementations from the OpenCV library on synthetic data. The experiment showed that the proposed approach has good performance in terms of accuracy and execution speed.

Keywords: visual odometry, optimization algorithm, reprojection error.

УДК 004.02

ББК 32.97

*Статья представлена к публикации
членом редакционной коллегии В.А. Уткиным.*

Поступила в редакцию 01.04.2025.

Опубликована 30.09.2025.