

Лобив И.В. Мехонцев Д.Ю. Мурзин Ф.А.

**ВОССТАНОВЛЕНИЕ ПОЛОЖЕНИЯ ТЕЛА В ПРОСТРАНСТВЕ
ПО ДАННЫМ, ПОСТУПАЮЩИМ С ОДНОМЕРНЫХ КАМЕР
ДЛЯ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ
АНАЛИЗА ДВИЖЕНИЯ ОБЪЕКТОВ**

ПОСТАНОВКА ЗАДАЧИ

В пространстве имеется твердое тело с известной геометрией, на котором в определенных местах закреплены светоизлучающие диоды, в дальнейшем называемые маркерами, которым присвоены уникальные номера. Также имеется набор одномерных детекторов, расположенных вокруг твердого тела, которые достаточно часто (200–300 раз в сек.) фиксируют положение маркеров. Каждый маркер обладает уникальными характеристиками (цвет, частота мерцания, и т.п.), по которым детектор может отличить его от других маркеров и восстановить его номер. В силу одномерности детектора, о каждом маркере, попадающем в его поле видимости, можно получить лишь уравнение плоскости, в которой этот маркер содержится. Все плоскости, получаемые от фиксированного детектора, пересекаются по одной прямой (“фокусная прямая”).

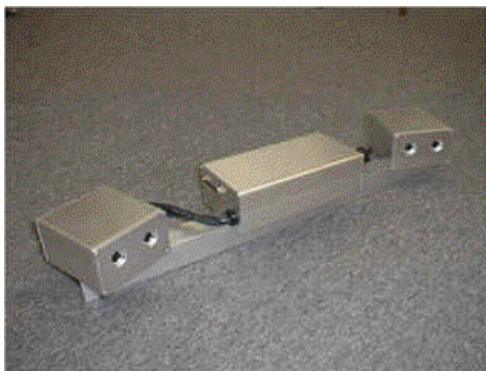


Рис. 1. Блок из четырех одномерных камер

Задача состоит в том, чтобы по имеющимся данным в реальном времени восстанавливать положение твердого тела с максимальной точностью.

На рис.1 показан блок из четырех одномерных камер. Таких блоков, расположенных в разных частях пространства, может быть несколько.

Осложняющие факторы

- В нескольких последовательных кадрах может быть недостаточно информации для однозначного восстановления положения тела (недоопределенность).
- В силу несовершенства детектора возможна ситуация, когда он выдает ошибочный номер маркера.

Математическая постановка

Положение тела в пространстве можно представить в виде $Ax + b$, где $b = (u, v, w)$ — вектор сдвига, A — ортогональная матрица поворота тела относительно начального положения.

Задача будет решена, если будут найдены величины A, b , при которых отклонение положения маркеров от соответствующих плоскостей будут минимальны в некоторой норме.

Научная новизна

Абсолютным лидером в данной области является фирма Vicon [5]. Ее продукты использовались для съемок таких знаменитых фильмов, как: «Titanic», «Gladiator», «The Mummy», «The Mummy Return», «Star Wars Episode 1 — the Phantom Menace», «The Patriot», «Enemy at the Gates», «Pearl Harbor» и др. Также ее продукты используются для постановки физических экспериментов, производства игр, обучения.

Рис. 2, 3 иллюстрируют работу продуктов разных фирм.

Попробуем выявить одну из главных проблем, которая возникает у исследователей [5, 6] в данной области. Подсчитаем объем информации, который должен обрабатываться в реальном времени.

Рассмотрим профессиональную 2d камеру с обычными характеристиками (1024x1024 True Color). Требование на рабочую частоту камеры — 300 фреймов в секунду (fps). Это требование вытекает из особенности предметной области. Частота промышленных мониторов и телеприемников — 24–30 fps, т.е. для того чтобы итоговое изображение на экране было непре-

рывным, частота на этапе обработки должна быть в 10 раз больше, т. е. 300 fps. Получается видеопоток 900Mb в секунду с каждой камеры. Поскольку камер 4 (например, в одной из конфигураций системы Vicon 8i), то в итоге получается более **3Gb** в секунду.



Рис. 2. Вид продукта лидера в данной области фирмы Vicon



Рис. 3. Сцена для работы продукта фирмы www.motionanalysisstudios.com

Этот показатель является огромным для сегодняшнего уровня развития вычислительной техники как для передачи такого объема информации в реальном времени, так и для обработки. Поэтому все направления исследований направлены на уменьшение размера информационного потока. Например, применение методов сжатия, увеличение количества вычислительных узлов и их мощности, привлечение суперкомпьютеров.

Мы предлагаем принципиально новое решение этого вопроса. Вычислим то же самое для одномерных камер. Размер изображения средней камеры 1024x1, условие на частоту то же самое 300 fps. Получаем с одной камеры 900Kb в секунду.

Так как камер требуется больше, то возьмем порядка 40 камер. Итого около **40Mb** в секунду, этот показатель уже вполне реален как для обработки, так и для передачи.

ОПИСАНИЕ МЕТОДА

Основная идея алгоритма заключается в том, чтобы в начале отфильтровать входные данные, а затем по ним восстановить положение тела [7]. При этом недостающая информация о положении тела дополняется данными с предыдущих кадров на основе интерполяции.

1. Случай, когда на одном временном шаге данных (плоскостей) достаточно для однозначного восстановления положения тела

Матрица A задается направлением (p, q, r) , относительно которого происходит поворот тела, и синусом угла поворота $\sin \phi$. Заметим, что вектор (p, q, r) одновременно может задавать и угол поворота по формуле $p^2 + q^2 + r^2 = \sin^2 \phi$, (мы используем модуль вектора для характеристики угла). Таким образом, матрица A задается тремя числами, и это представление есть непрерывное соответствие между ортогональными матрицами и единичным шаром в трехмерном пространстве.

Все это позволяет нам для нахождения преобразования $Ax + b$ воспользоваться методом наименьших квадратов, который в случае достаточности данных позволяет сразу по исходным данным (уравнениям плоскостей) восстановить положение тела, минуя определение положения каждого маркера по отдельности (что зачастую невозможно, потому что на каждый маркер может приходиться менее чем три плоскости).

Например, в случае если на одном шаге времени видны одновременно 3 маркера для восстановления положения тела, достаточно знать по две плоскости на каждый маркер (итого 6 плоскостей) или 1 плоскость про первый маркер, 2 плоскости про второй, 3 плоскости про третий маркер (также 6 плоскостей).

Метод наименьших квадратов позволяет также использовать избыточное количество плоскостей (более 6) для повышения точности восстановления. Пример показан на рис. 4.

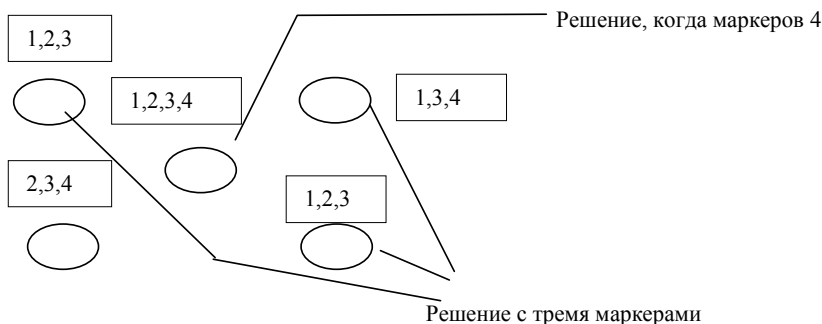


Рис. 4. Эффект увеличения точности восстановления положения при использовании информации с большего количества детекторов

Выпишем в явном виде представление матрицы A :

$$A = A(p, q, r) = \begin{pmatrix} c & -r & q \\ r & c & -p \\ -q & p & c \end{pmatrix} + \frac{1}{1+c} \begin{pmatrix} p^2 & pq & pr \\ qp & q^2 & qr \\ rp & rq & r^2 \end{pmatrix},$$

где $c^2 = 1 - (p^2 + q^2 + r^2) = \cos^2 \phi$.

Пусть на данном временном шаге известны N плоскостей, заданных в виде $(n_i, X) = d_i$ $i = 1 \dots N$, где $n_i = (a_i, b_i, c_i)$ — вектор нормали к плоскости, (\circ, \circ) — скалярное произведение, а также N точек $X_i = (x_i, y_i, z_i)$, характеризующих координаты маркера в системе координат твердого тела (предполагается, что эта информация считывается из файла с описанием твердого тела).

Метод наименьших квадратов приводит к следующей задаче:

$$\sum_{i=1}^N [(n_i, AX_i + b) - d_i]^2 \rightarrow \min,$$

Взяв частные производные соответственно по p, q, r, u, v, w , приходим к следующей системе уравнений:

$$\left\{ \begin{array}{l} \sum_{i=1}^N \left(\frac{\partial A}{\partial p} X_i, n_i \right) ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N \left(\frac{\partial A}{\partial q} X_i, n_i \right) ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N \left(\frac{\partial A}{\partial r} X_i, n_i \right) ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N a_i ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N b_i ((AX_i + b, n_i) - d_i) = 0 \\ \sum_{i=1}^N c_i ((AX_i + b, n_i) - d_i) = 0. \end{array} \right.$$

В полученную систему неизвестные u, v, w входят линейно, поэтому мы сразу можем их исключить (выразить через p, q, r) линейной заменой:

$$b = S^{-1} \sum_{i=1}^N d_i n_i - \sum_{i=1}^N S^{-1} S_i A X_i, \text{ где}$$

$$S_i = \begin{pmatrix} a_i^2 & a_i b_i & a_i c_i \\ b_i a_i & b_i^2 & b_i c_i \\ c_i a_i & c_i b_i & c_i^2 \end{pmatrix}, \quad S = S_1 + S_2 + \dots + S_N.$$

В итоге у нас остается система трех нелинейных уравнений с тремя неизвестными p, q, r :

$$\begin{cases} f_1(p, q, r) = 0 \\ f_2(p, q, r) = 0 \\ f_3(p, q, r) = 0 \end{cases}$$

или в векторном виде

$$F(P) = 0, \text{ где } F = (f_1, f_2, f_3), \quad P = (p, q, r).$$

Для ее решения воспользуемся методом последовательных приближений Ньютона [3].

Пусть на n -ом шаге известно приближение $P_n = (p_n, q_n, r_n)$, тогда следующее приближение P_{n+1} находится по формуле: $P_{n+1} = P_n - J^{-1}F(P_n)$, где J — матрица Якоби преобразования F ,

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial p} & \frac{\partial f_1}{\partial q} & \frac{\partial f_1}{\partial r} \\ \frac{\partial f_2}{\partial p} & \frac{\partial f_2}{\partial q} & \frac{\partial f_2}{\partial r} \\ \frac{\partial f_3}{\partial p} & \frac{\partial f_3}{\partial q} & \frac{\partial f_3}{\partial r} \end{pmatrix}.$$

Как известно, метод Ньютона очень быстро сходится в случае хорошего начального приближения.

В нашем алгоритме для начального приближения используются данные, полученные в результате экстраполяции по предыдущим временным шагам (с предыдущих кадров) — это обеспечивает хорошее начальное приближение и позволяет обойтись в методе Ньютона двумя-тремя итерациями для достижения нужной нам точности.

2. Случай, когда на одном временном шаге нам приходит недостаточное количество плоскостей (менее 6), для того чтобы однозначно восстановить положение тела

Очевидно, что в этом случае искомые величины p, q, r, u, v, w также удовлетворяют раннее описанной системе уравнений $F(P) = 0$, но решений эта система имеет много (как правило, имеется целое пространство решений).

Недостающие плоскости мы получаем с помощью экстраполяции по предыдущим временным шагам следующим способом.

На первом этапе вычисляются положения невидимых в данный момент точек на основе того, что мы знали их положение (а следовательно, приближенно знали вектор скорости и ускорения) на предыдущих шагах [4].

На втором этапе через вычисленные точки проводятся плоскости, которыми и дополняются исходные неполные данные.

Более точно, пусть (в таких же обозначениях, как и ранее) на данном временном шаге известны M дополнительных (полученных на основе интерполяции) плоскостей, заданных в виде $(n_i, X) = d_i$, $i = N + 1 \dots N + M$, а также M им соответствующих точек $X_i = (x_i, y_i, z_i)$.

Тогда в предположении, что $M + N \geq 6$, возникает задача [8]

$$\sum_{i=1}^{N+M} [(n_i, AX_i + b) - d_i]^2 \rightarrow \min,$$

полностью аналогичная предыдущему случаю.

Заметим, что этим методом можно вычислять более точное положение тела на предыдущих временных шагах, используя информацию, полученную на последующих шагах (направить метод в “прошлое”).

3. Фильтрация шумов

В силу несовершенства детекторов иногда фиксируются неверные плоскости (т.е. маркер не лежит в плоскости, а находится вдали от нее), например, если перепутается номер маркера.

Для борьбы с таким видом шумов положение тела восстанавливается дважды. Первый раз восстановление происходит по всем имеющимся плоскостям (в том числе и неверным).

Затем мы проверяем, насколько каждая плоскость отклоняется от вычисленного положения, и при втором восстановлении не учитываются те плоскости, в которых это отклонение велико (например, в 3 раза превосходит среднее отклонение и т.п.).

ОПИСАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

Была написана реализация вышеизложенного алгоритма на языке C++ [1,2]. В виде dll-библиотеки для Windows [9,10] и в виде запускаемого файла для Linux. Программа тестировалась на данных, полученных с реальной системы, и показала высокую точность и устойчивость к шумам. Среднее время обработки — 10000 временных шагов менее 1 сек.

СПИСОК ЛИТЕРАТУРЫ

1. **Gamma E., Helm R., Johnson R., Vlissides J.** Design Patterns. — Addison Wesley Longman Inc, 1995 — 132 p.
2. **Booch G.** Object-oriented analysis and design. 2-nd ed. — Addison Wesley Longman Inc, 1994 — 187 p.
3. **Самарский А.А.** Введение в численные методы. — М.: Наука, 1982. — 272 с.
4. **Рябенский В.С.** Введение в вычислительную математику. Учебное пособие для вузов. — М.: Физматлит, 1994. — 336 с.

5. **Vicon Inc.** <http://www.vicon.com>
6. **Motion Analysis Studio Inc.** <http://www.motionanalysisstudios.com>
7. **Мехонтцев Д.Ю., Лобив И.В., Селезнев К.С.** Слежение и определение скорости движущихся на плоскости объектов в реальном времени. // Современные проблемы конструирования программ. — Новосибирск, 2002. — С. 243–246.
8. **Коновалов А.Н.** Введение в вычислительные методы линейной алгебры. — Новосибирск: ВО «Наука», 1993. — 59 с.
9. **Codeguru** <http://www.codeguru.com>
10. **Code Project** <http://www.codeproject.com>