



УДК 004.272:004.942

ББК 32.973-018.2; 32.81; 28.071

АСПЕКТЫ ИСПОЛЬЗОВАНИЯ ПАКЕТА MATLAB НА ВЫЧИСЛИТЕЛЬНОМ КЛАСТЕРЕ ДЛЯ РЕШЕНИЯ БИОМЕТРИЧЕСКИХ ЗАДАЧ

А.В. Карпов, О.В. Комогорцев

В статье рассматриваются особенности использования пакета MATLAB для проведения математического моделирования на вычислительном кластере Университета штата Техас (Texas State University, San Marcos, Texas). Решаемая задача заключается в исследовании возможности идентификации человека с помощью биометрической информации. Ее источником служат глаза человека. По предварительно записанным движениям глаз подопытных определяются параметры математической модели глаза, позволяющие наиболее точно смоделировать записанную траекторию. Наборы таких параметров используются в дальнейшем для идентификации. Для определения значения параметров используются методы математической оптимизации. В силу большого количества данных этот процесс требует значительного процессорного времени. Для ускорения решения была написана параллельная программа на языке среды MATLAB. В ходе ее разработки были выявлены некоторые особенности библиотеки Parallel Toolbox пакета, которые необходимо учитывать при использовании MATLAB на вычислительном кластере.

Ключевые слова: биометрика, биоинформатика, параллельное программирование, математическая модель глаза, MATLAB, вычислительный кластер.

Введение

В современном мире задача аутентификации пользователя достаточно важна. Вплоть до настоящего момента наиболее популярным методом ее решения является ввод логина и пароля пользователя. После успешного ввода пароля, PIN-кода, использования смарт-карты и иных аналогов пользователь получает доступ. К сожалению, пароль и логин относительно легко могут узнать злоумышленники. Поэтому требуется достоверный метод подтверждения личности владельца учетной записи.

В качестве такого метода чаще всего выбираются методы биометрической идентификации. К популярным метрикам, среди прочих, относятся отпечатки пальцев [6], радужка глаза [2; 5], лицо [22] и отпечатки ладони [17]. В общем случае от методов биометрической идентификации требуется неинвазивность и работа в реальном режиме времени. Однако такие методы уязвимы для злоумышленников, поскольку современные технологии позволяют с допустимой погрешностью воспроизвести проверяемые параметры [18–21]. Кроме того, известны случаи, когда органы пользователя хирургически извлекались для обхода систем безопасности [8].

Поэтому определенный интерес представляет источник биометрической информации, отвечающий следующим требованиям: позволяет выполнять аутентификацию пользователя в режиме реального времени либо близком к нему; достаточно защищен от подделок; при извлечении из человека меняет свои свойства в мере, достаточной для отказа в аутентификации.

Одним из потенциальных кандидатов является человеческий глаз. Он служит источником множества биометрических данных, таких как радужка глаза, сетчатка глаза, размер зрачка, характер и скорость движения глаз. В рамках данной статьи рассматривается одна метрика — траектории движения глаз человека.

1. Математическая модель глаза человека

Упрощенно систему зрения человека можно представить в виде глазного яблока, управляющих мышц, нервов для передачи сигналов на мышцы и мозг человека для генерации управляющих сигналов [3]. Эту упрощенную систему можно представить в виде механической модели, состоящей из глазного яблока, тяг и пружин, управляемой сигналами аналога нервной системы [1]. В свою очередь, механическая модель и контрольное устройство описываются системой линейных дифференциальных уравнений:

$$\left\{ \begin{array}{l} \dot{x}_1(t) - x_4(t) = 0, \\ \dot{x}_2(t) - R_1 R_2 x_1(t) + R_2 x_2(t) - \frac{R_1}{R_4} x_5(t) = 0, \\ \dot{x}_3(t) - R_1 R_3 x_1(t) + R_3 x_3(t) - \frac{R_1}{R_5} x_6(t) = 0, \\ \dot{x}_4(t) + R_6 x_1(t) - R_7 x_2(t) - R_7 x_3(t) + R_8 x_4(t) = 0, \\ \dot{x}_5(t) + \frac{1}{\tau_{ag}} x_5(t) - \frac{N_{LR}}{\tau_{ag}} = 0, \\ \dot{x}_6(t) + \frac{1}{\tau_{ant}} x_6(t) - \frac{N_{MR}}{\tau_{ant}} = 0. \end{array} \right. \quad (1)$$

Коэффициенты $R_1 = \frac{K_{SE}}{K_{SE} + K_{LT}}$, $R_2 = \frac{K_{SE}}{\hat{B}_{AG}}$, $R_3 = \frac{K_{SE}}{\hat{B}_{ANT}}$, $R_4 = \hat{B}_{AG}$, $R_5 = \hat{B}_{ANT}$, $R_6 = \frac{2K_{SE} + K_P}{J}$, $R_7 = \frac{K_{SE}}{J}$, $R_8 = \frac{B_P}{J}$ выражаются через параметры модели глаза. Параметры K_{SE} , K_{LT} , K_P , B_P , \hat{B}_{AG} , \hat{B}_{ANT} и J относятся к физической части модели (описывают коэффициенты сжатия, растяжения, упругости, инерцию вращения и т. п.) и являются константами. Однако $N_{LR}(t, \Theta)$, $N_{MR}(t, \Theta)$, $\tau_{ag}(t, \Theta)$ и $\tau_{ant}(t, \Theta)$ описывают сигналы управления физической моделью. В простейшем случае это кусочно-линейные функции от времени t и моделируемого угла поворота глаза Θ . В результате аналитическое решение системы (1) хотя теоретически и возможно, но в настоящий момент не

получено. Подробнее с использованной моделью можно ознакомиться в работах [9; 10].

Модель (1) позволяет воспроизводить все возможные типы движения глаза человека. Однако в рамках данного исследования использовались только фиксации и саккады. Фиксации — это фокусирование глаза на неподвижной цели, саккады — это движение глаз, вызываемое мгновенным перемещением цели из одного положения в другое [3].

Саккады использовались для определения параметров данной модели, свойственных подопытным. Этими параметрами считаются параметры модели, обеспечивающие минимальное различие между записанной и смоделированной траекториями. Для их нахождения решалась задача математической оптимизации с целевой функцией $f = \sum_{i=1}^n |r_i - s_i|$, где n — длина траектории; r_i — координаты записанного положения глаза в момент времени i , а s_i — координаты смоделированного положения глаза, полученные путем численного решения системы (1). Оптимизация проводилась встроенным оптимизатором MATLAB `fminsearch` по параметрам модели. При решении задачи для одной саккады выполнялось в среднем порядка 30 000 численных решений системы (1).

Для упрощения автоматической классификации записанных движений глаз саккады разделялись фиксациями. Алгоритмы классификации и особенности их реализации описаны в работах [7; 11; 12].

Наборы этих параметров, определенные для всех записанных саккад подопытных, в дальнейшем используются для идентификации. Методика идентификации и результаты ее применения доступны в [4; 13–15].

2. Численное решение

2.1. Параллельная программа

В используемой базе 236 файлов и около 24 000 саккад. Среднее время определения параметров модели для одной саккады с помощью оптимизированной однопоточной программы в среде MATLAB составляет порядка 15 минут при условии выполнения программы на рабочей станции Dell Optiplex 780 (ЦП Intel Core2 Quad Q9400 2.6ГГц, 8Гб памяти). Ожидаемое время обработки всех записей на одной машине примерно год.

Поскольку саккады независимы друг от друга, эта задача обладает естественным параллелизмом. Поэтому запуск нескольких независимых копий программы позволяет почти линейно сокращать время работы. Использование всех доступных машин уменьшило это время до трех недель. Однако такой метод требует много ручной работы по подготовке и копированию данных на машины, запуску программ и сбору готовых результатов. С учетом того, что: после каждого изменения параметров модели и кода требуется провести расчет заново; остальные пользователи регулярно останавливали счет; общее число записей больше доступных в базе на порядок, естественным было реализовать версию программы для работы на кластере.

Современные версии MATLAB позволяют использовать три метода организации параллельных программ. Наиболее простым способом является использование пула процессов-исполнителей (`workers`) MATLAB (`matlabpool`). После инициализации пула становится возможным использование конструкции итеративного параллелизма `parfor`. Однако использование данного метода требует определенной организации ЛВС, которая в нашем случае невозможна из-за политики сетевой безопасности университета. Другим вариантом является использование функций MPI. Однако политика лаборатории требует использования только MATLAB, и поэтому данный метод в конкретных обстоятельствах

был признан нецелесообразным.

Третий вариант — использование механизма заданий MATLAB. Он и был выбран для использования в параллельной версии программы. При этом способе организации работы с кластером MATLAB формирует пакет задач — так называемое задание. Оно передается планировщику, который распределяет полученные задачи задания по узлам кластера. По завершении работы всех задач планировщик собирает результаты работы со всех узлов кластера и возвращает их пакету.

Собственно задания были реализованы как объекты MATLAB, в свойствах которых хранились все необходимые данные, а поиск параметров модели выполнялся вызовом метода этих объектов. Такая организация позволила достичь хорошей масштабируемости программы, а также с минимумом изменений использовать одну программу как на кластере, так и на локальном компьютере с различным числом ядер.

2.2. Вычислительный кластер

Кластерная версия пакета MATLAB 2009a установлена на кластер, состоящий из 30 узлов. Каждый узел содержит два четырехядерных процессора Intel Xeon E5540 с тактовой частотой 2,5 ГГц и 24 Гб памяти. HyperThreading отключен. На двух узлах установлены Nvidia Tesla для ускорения расчетов с использованием технологии CUDA от NVidia. В качестве управляющей сети используется сеть Gigabit Ethernet, среда обмена данными — Infiniband. Узлы подключены к накопителям с помощью протокола iSCSI, транспортной средой служит Gigabit Ethernet с аппаратной поддержкой объединения четырех портов в один. В качестве системы управления заданиями на кластере используется планировщик LSF фирмы Platform, поддерживающий как MPI-программы, так и MATLAB-задания.

Имеющиеся лицензии MATLAB позволяют использовать одновременно только 64 узла из 320 доступных, что ограничивает количество параллельно обрабатываемых саккад.

2.3. Экспериментальные данные

В экспериментах приняло участие 59 испытуемых. Для записи траекторий использовался айтрекер EyeLinkII компании SR Research. Запись осуществлялась с частотой 1 ГГц. В качестве стимула использовались два типа саккад — вертикальные и горизонтальные. Вертикальные саккады генерировались с амплитудой 20° экрана. Запись содержала 50 циклов — саккада снизу вверх, фиксация и последующая саккада сверху вниз. Горизонтальные саккады были разбиты на две группы. У 27 испытуемых амплитуда горизонтальных саккад составляла 20°, а у 32 — 30°. Оба типа стимулов также состояли из 50 циклов — левонаправленная саккада, фиксация и правонаправленная саккада. Каждый испытуемый записывался дважды с промежутком в 20 минут между сессиями. Каждая сессия состояла из записи горизонтальных и вертикальных саккад. База записей [16] условно свободна, подробнее она описана в работах [4; 13].

Заключение

Разработанная параллельная программа позволяет определить параметры для базы записей на использовавшемся кластере примерно за 48 часов без постоянного контроля

со стороны оператора. Это позволяет оперативно проверять изменения в алгоритмах и моделях. Результаты работы программы были использованы в работах [4; 13; 15; 16]. Следует учесть, что особенностью решаемой задачи является естественное разбиение на независимые друг от друга подзадачи. Это позволило достичь высокой эффективности использования MATLAB 2009a в организации параллельных вычислений на кластере.

Однако планировщик, являющийся неотъемлемой частью использованного метода распараллеливания, обладает определенными недостатками. А именно:

- относительно высокие накладные расходы (в среднем 2 минуты на постановку задания на выполнение при незагруженном кластере и сбор результатов) в использованной версии 2009a (в марте 2012 г. в ходе плановой профилактики MATLAB был обновлен до версии 2012a, в которой данная задержка была устранена);
- невозможность запуска нескольких заданий одновременно, поскольку планировщик не ведет учета используемых в данный момент лицензий;
- атомарность задания — невозможно узнать о состоянии задач задания во время его выполнения на кластере;
- отсутствие взаимодействия задач между собой в рамках одного задания;
- как следствие, время выполнения задания равно времени выполнения самой долгой задачи из задания.

Эти особенности необходимо учитывать при принятии решения об использовании MATLAB с планировщиком заданий для организации параллельных вычислений на кластере. Очевидно, что задания MATLAB наиболее эффективны в случае возможности разбиения задачи на подзадачи. При этом каждая подзадача должна считаться относительно долго, не зависеть от других подзадач, и все подзадачи должны выполняться за примерно одинаковое время. Также эти ограничения должны учитываться на стадии разработки алгоритма программы.

Кроме того, при обновлении версии MATLAB до 2012a в марте 2012 г. произошло изменение функций распараллеливания, что привело к необходимости внести соответствующие изменения в программу.

СПИСОК ЛИТЕРАТУРЫ

1. Bahill, A. T. Development, Validation and Sensitivity Analyses of Human Eye Movements Models / A. T. Bahill // CRC Critical Reviews in Bioengineering. — 1980. — V. 4. — P. 311–355.
2. Daugman, J. G. High Confidence Visual Recognition of Persons by a Test of Statistical Independence / J. G. Daugman // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1993. — V. 15. — P. 1148–1161.
3. Duchowski, A. T. Eye Tracking Methodology: Theory and Practice / A. T. Duchowski. — London : Springer-Verlag, 2003. — 328 p.
4. Holland, C. Biometric Identification via Eye Movement Scanpaths in Reading / C. Holland, O. V. Komogortsev // In Proceedings of the IEEE International Joint Conference on Biometrics (IJCB). — 2011. — P. 1–8.

5. Hollingsworth, K. All Iris Code Bits are Not Created Equal / K. Hollingsworth, K. W. Bowyer, P. J. Flynn // *Biometrics: Theory, Applications, and Systems. BTAS 2007*. — 2007. — P. 1–6.
6. Jain, A. A Multimodal Biometric System Using Fingerprint, Face and Speech / A. Jain // *Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*. — 1999. — P. 182–187.
7. Karpov, A. V. Automated Classification and Scoring of Smooth Pursuit Eye Movements in Presence of Fixations and Saccades / A. V. Karpov, O. V. Komogortsev // *Journal of Behavioral Research Methods*. — 2012 (in press).
8. Kent, J. Malaysia Car Thieves Steal Finger / J. Kent // *BBC Online*. — 2005. — Mode of access: <http://news.bbc.co.uk/2/hi/asia-pacific/4396831.stm>.
9. Komogortsev, O. 2D Oculomotor Plant Mathematical Model for Eye Movement Simulation / O. Komogortsev, U. Jayarathna // *In Proceedings of the 8th IEEE International Conference on Bioinformatics and Bioengineering (BIBE)*. — 2008. — P. 1–8.
10. Komogortsev, O. Eye Movement Prediction by Kalman Filter with Integrated Linear Horizontal Oculomotor Plant Mechanical Model / O. Komogortsev, J. Khan // *In Proceedings of the Eye Tracking Research & Applications Symposium (ETRA 2008)*. — 2008. — P. 229–236.
11. Komogortsev, O. V. Qualitative and Quantitative Scoring and Evaluation of the Eye Movement Classification Algorithms / O. V. Komogortsev, U. K. S. Jayarathna, D. H. Koh, M. Gowda // *In Proceedings of ACM Eye Tracking Research & Applications Symposium*. — Austin, TX, USA, 2010. — P. 1–4.
12. Komogortsev, O. V. Standardization of Automated Analyses of Oculomotor Fixation and Saccadic Behaviors / O. V. Komogortsev, D. V. Gobert, U. K. S. Jayarathna, D. H. Koh, M. Gowda // *IEEE Transactions on Biomedical Engineering*. — 2010. — V. 57, № 11. — P. 2635–2645.
13. Komogortsev, O. V. Biometric Authentication via Oculomotor Plant Characteristic / O. V. Komogortsev, A. V. Karpov, L. Price, C. Aragon // *In Proceedings of the IEEE/IARP International Conference on Biometrics (ICB)*. — 2012. — P. 1–8.
14. Komogortsev, O. V. Biometric Identification via an Oculomotor Plant Mathematical Model / O. V. Komogortsev, U. K. S. Jayarathna, C. R. Aragon, M. Mechehoul // *In Proceedings of ACM Eye Tracking Research & Applications Symposium*. — Austin, TX, USA, 2010. — P. 1–4.
15. Komogortsev, O. V. Biometric Authentication via Anatomical Characteristics of the Oculomotor Plant Characteristic / O. V. Komogortsev, A. Karpov, C. Aragon // *In Proceedings of the IEEE/IARP International Conference on Biometrics (ICB)*. — 2012. — P. 1–8.
16. Komogortsev, O. V. Eye Movement Biometric Database (EMBD) v1 / O. V. Komogortsev, A. V. Karpov. — Electronic text data. — Mode of access: http://cs.txstate.edu/~ok11/embd_v1.html. — Title from screen.
17. Kong, A. A Survey of Palmprint Recognition / A. Kong, D. Zhang, M. Kamel // *Pattern Recognition*. — 2009. — V. 42. — P. 1408–1418.
18. Puhan, N. B. Iris Liveness Detection for Semi-transparent Contact Lens Spoofing / N. B. Puhan, S. Natarajan, A. S. Hegde // *Communications in Computer and Information Science (CCIS)*. — 2011. — V. 205. — P. 249–256.
19. Roberts, C. Biometric Attack Vectors and Defences / C. Roberts // *Computers & Security*. — 2007. — V. 26. — P. 14–25.

20. Ruiz-Albacete, V. Direct Attacks Using Fake Images in Iris Verification / V. Ruiz-Albacete, P. Tome-Gonzalez, F. Alonso-Fernandez, J. Galbally, J. Fierrez, J. Ortega-Garcia // BIOD. — 2008. — P. 181–190.
21. Williams, J. M. Biometrics or ... Biohazards? / Williams J. M. // Proceedings of the 2002 workshop on New Security Paradigms. — Virginia Beach, Virginia, 2002. — P. 97–107.
22. Wiskott, L. Face Recognition by Elastic Bunch Graph Matching / L. Wiskott, J.-M. Fellous, N. Kruger, C. von Malsburg // IEEE Transactions on Pattern Analysis and Machine Intelligence. — 1997. — V. 19, № 7. — P. 775–779.

ASPECTS OF MATLAB USAGE ON COMPUTATIONAL CLUSTER FOR SOLVING OF BIOMETRICS PROBLEMS

A.V. Karpov, O.V. Komogortsev

The paper considers the particular use of MATLAB for mathematical modeling on computational cluster at the Texas State University — San Marcos. The considered problem is to study the possibility of person identification based on biometric information from human eye. From recorded trajectories parameters of oculomotor plant were extracted. These parameters should produce simulated trajectories with minimal difference from actual ones. This extraction is a solving of mathematical optimization problem. Due to a large amount of data this requires significant CPU time. To speed up the estimation of parameters a parallel program was written in the MATLAB language.

Key words: *biometrics, bioinformatics, parallel programming, oculomotor plant, MATLAB, computational cluster.*