

PARALLELIZATION OF THE MATHEMATICAL MODEL OF GABOR FILTER

Serghii Rassomakhin, Artur Bykov

V. N. Karazin Kharkiv National University, 6 Svobody Sq., Kharkiv, 61022, Ukraine
rassomakhin@karazin.ua, bykov.artur.kh.96@gmail.com

Reviewer: Vyacheslav Kalashnikov, Doctor of Sciences (Physics and Mathematics), Full Prof., Department of Systems and Industrial Engineering, Campus Monterrey, Monterrey, 64849, Mexico.
kalash@itesm.mx

Received: September 2019

Abstract: Due to the acceleration of the informatization of modern society and the increase in the number of flows and objects of information that must be protected from unauthorized access, problems with the use of biometric identification technologies to differentiate access to information resources are becoming increasingly important. The paper discusses some features of fingerprint preprocessing procedures using the Gabor filter. A block diagram of a parallel model of software for fingerprint preprocessing is presented. Various technologies are also briefly considered that make it possible to organize an appropriate program (parallel processing), the advantages and disadvantages of each of them are described, on the basis of which the optimal one is selected for a specific task. The results on the speed of program execution with a different number of threads and their priority are provided.

Key words: fingerprints; Biometric Image; Gabor Function; Parallel Execution

1 Introduction

Due to the increasing informatization of modern society and the increasing number of flows and objects of information that need to be protected from unauthorized access, problems with the use of biometric identity identification technologies to differentiate access to information resources are becoming more urgent. The use of biometric characteristics to confirm an identity involves the use of physical characteristics, such as voice or fingerprints, for identification purposes. Fingerprint mapping is the most successful biometric identification technology due to its ease of use, non-interference and reliability. The fingerprint consists of furrows and stripes, forming a complex pattern unique to each person, and therefore provide the optimum method of verification. Usually, the problem with fingerprint identification, namely fingerprints, is poor image or fuzzy lines. The subject matter of the development is an information and technical system for the processing of fingerprint images. The problem they solve is that it is necessary to improve the image of fingerprints for further analysis of fingerprints.

2 The Gabor filter and implementation of the software product

The application of the Gabor filter enhances the image. It is contrasted with the lines of prints, noise is removed. Binarized images are convenient for further conversion: skeletonization and search for special points on prints [1-6]. Using coherence allows you to highlight the area of interest in which additional calculations are performed. The disadvantage of the algorithm is the representation of a curve line as a set of directions in each segment, which can cause a fuzzy response of the filter in the areas and curvatures and curls. On the other hand, the incremental sampling associated with the decrease in segment size entails a large margin of error in determining the average line orientation of each segment.

Gabor filter transfer function

$$G(x, y, \varphi) = e^{\left[\frac{-\frac{1}{2}(x \cos(\varphi) + y \sin(\varphi))^2}{\sigma_x^2} + \frac{(-x \sin(\varphi) + y \cos(\varphi))^2}{\sigma_y^2} \right]} \cos(2\pi\theta(x \cos(\varphi) + y \sin(\varphi))) - Z.. \quad (1)$$

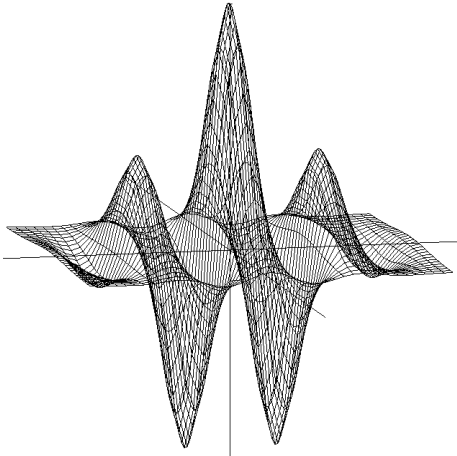


Fig. 1 – Gabor function

The C⁺⁺ programming language was chosen to implement the software. C⁺⁺ is a compiled, statically typed general-purpose programming language. Supports programming paradigms such as procedural programming, object-oriented programming, generalized programming. The language has a rich standard library, which includes common containers and algorithms, I/O, regular expressions, lots of accuracy support, and other features. C⁺⁺ combines the properties of both high-level and low-level languages. The advantage of written software is the lack of third-party libraries. The program is based solely on standard tools provided by Microsoft. As well as minimizing the use of objects for easy translation of the program into the C programming language, as needed. The pro-

gram has no memory leaks and handlers (*repeated testing*).

In order to write a program that in turn will use the computing power of the computer as efficiently as possible. The main technologies for parallelizing software, namely the Message Passing Interface (MPI), OpenMP (Open Multi-Processing) and POSIX Threads, were reviewed. Each technology has a number of advantages and disadvantages.

MPI (message passing interface) - the transmission system is standardized (function library). The standard is the syntax and semantics of the current functions, use when writing portable programs with the transmission of translations on the moves Fortran 77, C and C⁺⁺. The main way of sharing parallel processes in a given whistle is to transfer a single site to a single site. The MPI standard is the interface, which is guilty of being impaired as a system by the program on the leather calculating platform, so we are corrupted when setting up our programs [7]. MPI-program - with no parallel processes. All processes are done once, asserting parallel part of the programs. Each process works in its address space, there are no common variables or data in MPI. The main way to interact with the processes ϵ is to make sure that one of the processes is completed.

It's independent of those that use MPI programs to show the highest level of productivity, the technology itself has a number of short-lived [7]:

- Low level (programming on the MPI is often the same as programming on the assembler), the need for detailed control of the output and the number of cycles, as well as the greater the number of processes — all the more important;
- Necessity of superlative specificity of types in tributes, as well as the very manifestation of hardcore divisions in tipi of transferred tributes;
- Flexibility of writing programs, keeping your eyes peeled at the end of the week of high-tech and high-quality processes - to rob is practically impractical to re-enter the wake-up MPI-programs;
- Visibility of the object-oriented approach.

OpenMP promotes a simple mechanism for realizing parallel parallels in additions for additional multithreading, in the “master” thread and the “slave” thread, and the redundancy of them. OpenMP standard for Fortran, C / C⁺⁺ (*Fortran is not to be looked at*) and for formulating an API for writing sterile bagging threading on multiprocessor systems using the Single Program Multiple Data (SPMD) [8]. Portability is tied to the OpenMP program model, which requires an independent platform for compiling directives (pragmas), functions and middleware, obviously showing a compiler, and I'm talking about parallelism. For a programmer, there's no need for additional storage, for getting things done, for synchronizing, for balancing and for getting things done. OpenMP allows you to see how it's possible over Pthreads, the POSIX interface for organizing threads. At Pthreads itself - it's a busy low level of programmability, a little bit of parallelism for giving, and the very

mechanism of flowing to bed is not for the purposes of organizing parallelism. OpenMP has terminology and programming model, close to Pthreads, for example, dynamically generated threads, external and collective data, etc. OpenMP technology is aimed at those who want to use one version of the program for a parallel and last weekend. However, it's possible to set programs so that you can correct ones in parallel mode, but give the last result in the last mode. In addition, through the accumulation of forgiveness rounded off, the result of the weekend shall be deducted from the cities of the Republic of Kazakhstan in possible deceases.

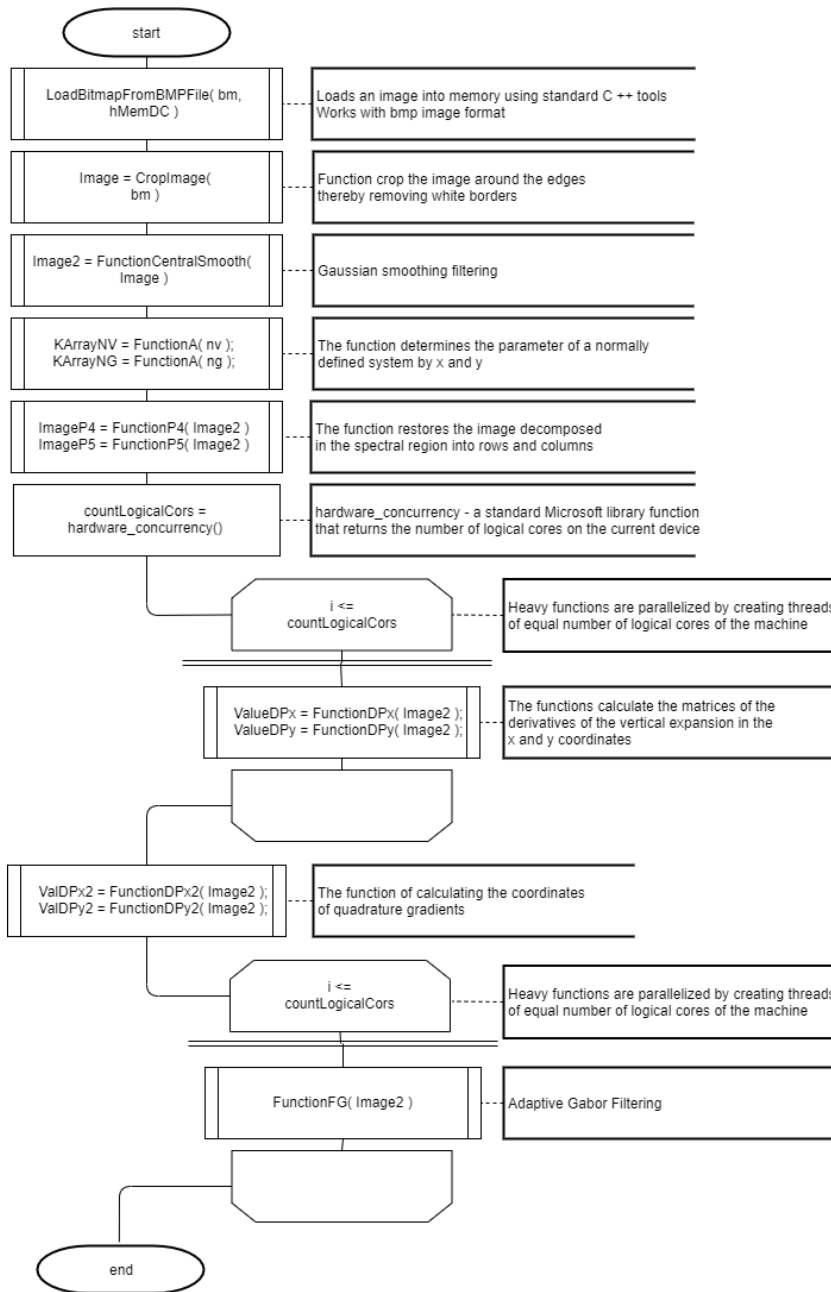


Fig. 2 – Block diagram

made it possible to fully test the speed of a program with a different number of threads.

Table. 1 – Program speed results

Thread Count / Priority	1	2	3	4
NORMAL	47s	39s	31s	28s
HIGHEST	45s	30s	23s	20s

POSIX Threads, usually referred to as pthreads, is an execution model that exists independently from a language, as well as a parallel execution model. It allows a program to control multiple different flows of work that overlap in time. Each flow of work is referred to as a thread, and creation and control over these flows is achieved by making calls to the POSIX Threads API. POSIX Threads is an API defined by the standard POSIX.1c, Threads extensions (IEEE Std 1003.1c-1995) [9].

Before performing the Gabor function on the image, a number of manipulations are necessary [1,3]. Some of them are quite complex and also need to be parallelized. Since the process of creating and starting the stream is quite heavy, the optimal solution for parallelizing the program was found. Figure 2 shows a block diagram of the main function which describes the main stages of the work progress.

3. Program execution speed testing

Product testing was conducted on a computer HP 4540s Intel i5 processor - 3230M CPU 2.6GHz. This processor model contains 4 logical cores, which

Figure 3 shows an example of an input image and Figure 4 shows an example of an image modified by a Gabor filter.



Fig.3 – Input image



Fig.4 – Modified image

3. Conclusions

A brief overview of the main procedures has been completed of fingerprint image preprocessing by the Gabor filter. Having also analyzed the most common technologies for organizing parallel execution of processes available in the C⁺⁺ programming language. Taking into account the fact that the mathematical model can be divided into subtasks that will not be interconnected, that is, the principles of data locality are preserved - we can conclude that there are no advantages for this task in the mpi technology. The next technology on the list is Open MP, in which the model is pretty close to our task (*processing of fingerprint image*), but it also has a number of disadvantages, the main one being the fact that the Microsoft C⁺⁺ compiler only supports ver. 2.0, which does not have support for setting priorities for child threads, which is significant limits our possibilities. Therefore, for a specific task, POSIX Threads is the preferred option that encapsulates fairly quick and convenient thread manipulations.

Посилання

- [1] Гудков В. Ю., Бойцов А. В. Улучшение изображений отпечатков пальцев с помощью фильтра Габора. URL: <https://cyberleninka.ru/article/n/uluchshenie-izobrazheniy-otpechatkov-paltsev-s-pomoschyu-filtra-gabora> (дата звернення 05.09.2019).
- [2] Костенко Л. С. Методы и алгоритмы сглаживания фона изображений в системах распознавания образов / Л. С. Костенко. Открытые информационные и компьютерные интегрированные технологии. 2014. Вып. 64. С. 177-181. URL: http://nbuv.gov.ua/UJRN/vikt_2014_64_21 (дата звернення 03.09.2019).
- [3] Ляхов П. А. Валуева М. В. Применение сглаживающих фильтров для очистки от шума изображений в оттенках серого. Наука. Инновации. Технологии. 2015. №3. URL: <https://cyberleninka.ru/article/n/primenenie-sglazhivayuschih-filtrov-dlya-ochistki-ot-shuma-izobrazheniy-v-ottenkah-serogo> (дата звернення 05.09.2019).
- [4] Федюкович Г. Г. Сравнение использования технологий параллельного программирования Microsoft применительно к задаче поиска данных MapReduce URL: <http://omega.sp.susu.ru/books/conference/PaVT2010/poster/169.pdf> (дата звернення 05.09.2019).
- [5] Гудков В. Ю., Бойцов А. В. Улучшение изображений отпечатков пальцев с помощью фильтра Габора. URL: <https://cyberleninka.ru/article/n/uluchshenie-izobrazheniy-otpechatkov-paltsev-s-pomoschyu-filtra-gabora> (дата звернення 05.09.2019).
- [6] Рыканов А.С. Анализ методов распознавания отпечатков пальца. Системы обработки информации. 2010. Вып.6 (87) . С.164 – 171. URL: http://www.hups.mil.gov.ua/periodic-app/article/7838/soi_2010_6_37.pdf (дата звернення 05.09.2019).
- [7] MPI: A Message-Passing Interface Standard Version 3.0. URL: <https://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf> (дата звернення 25.09.2019).
- [8] Параллельное программирование на OpenMP. URL: <http://ccfit.nsu.ru/arom/data/openmp.pdf> (дата звернення 11.10.2019).
- [9] Blaise V. POSIX Thread – Introduction to multithreading. URL: <https://computing.llnl.gov/tutorials/pthreads/> (дата звернення 07.09.2019).

Рецензент: В'ячеслав Калашников, д.ф.-м.н., проф., Технологический университет Монтеррея, пр. Еухенио Гарса Сада 2501, Монтеррей, 64849, Мексика.

E-mail: kalash@itesm.mx

Надійшло: Вересень 2019.

Автори:

Артур Биков, студент факультету комп'ютерних наук, ХНУ ім. В.Н. Каразіна, майдан Свободи 4, м. Харків, 61022, Україна.

E-mail: bykov.artur.kh.96@gmail.com

Сергій Рассомахін, д.т.н., проф., ХНУ ім. В. Н. Каразіна, майдан Свободи 6, Харків, 61022, Україна.

E-mail: rassomakhin@karazin.ua

Розпаралелювання математичної моделі фільтра Габора.

Анотація. У зв'язку з прискоренням інформатизації сучасного суспільства та збільшенням числа напрямків і об'єктів інформації, які необхідно захистити від несанкціонованого доступу, проблеми з використанням технологій біометричної ідентифікації особистості для розмежування доступу до інформаційних ресурсів стають дедалі актуальнішими. В роботі розглядаються деякі особливості процедур попередньої обробки відбитків пальців з використанням фільтра Габора. Представлена структурна схема паралельної моделі програмного забезпечення для попередньої обробки відбитків пальців. Також коротко розглянуті різні технології, що дозволяють організувати відповідну програму (паралельної обробки), описуються переваги і недоліки кожної з них, на основі яких вибирається оптимальна, для конкретного завдання. Надамо результати по швидкості виконання програм з різною кількістю потоків і їх пріоритетом.

Ключові слова: відбитки пальців; біометричне зображення; функція Габора; паралельне виконання.

Рецензент: Вячеслав Калашников, д.т.н., проф., Технологический университет Монтеррея, пр. Еухенио Гарса Сада 2501, Монтеррей, 64849, Мексика.

E-mail: kalash@itesm.mx

Поступила: Сентябрь 2019.

Авторы:

Артур Биков, студент факультета компьютерных наук, ХНУ им. В. Н. Каразина, пл. Свободы, 4, Харьков, 61022, Украина.

E-mail: bykov.artur.kh.96@gmail.com

Сергей Рассомахин, д.т.н., проф., ХНУ им. В. Н. Каразина, пл. Свободы 6, Харьков, 61022, Украина.

E-mail: rassomakhin@karazin.ua

Распараллеливание математической модели фильтра Габора.

Аннотация. В связи с ускорением информатизации современного общества и увеличением числа направлений и объектов информации, которые необходимо защитить от несанкционированного доступа, проблемы с использованием технологий биометрической идентификации личности для разграничения доступа к информационным ресурсам становятся все более актуальными. В работе рассматриваются некоторые особенности процедур предварительной обработки отпечатков пальцев с использованием фильтра Габора. Представлена структурная схема параллельной модели программного обеспечения для предварительной обработки отпечатков пальцев. Также коротко рассмотрены различные технологии, позволяющие организовать соответствующую программу (параллельной обработки), описываются преимущества и недостатки каждой из них, на основе которых выбирается оптимальная, для конкретной задачи. Предоставлены результаты по скорости выполнения программ с разным количеством потоков и их приоритетом.

Ключевые слова: отпечатки пальцев; биометрическое изображение; функция Габора; параллельное исполнение.