STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LVI**, Number 1, 2011

OPTICAL ERATOSTHENES SIEVE

LIVIU ŞTIRB

Abstract. The sieve of Eratosthenes is a simple algorithm for finding all prime numbers up to a specified natural number. Despite its simplicity it is not widely used because of its complexity: O($n$log$n$loglog$n$) bit operations. An optical implementation for the sieve of Eratosthenes is described in this paper. The proposed device is highly parallel, by doing multiple verifications in the same time, thus providing a significant speedup compared to one implemented on a single core electronic computer.

1. Introduction

Solving problems with optical devices has received an increasing attention in the recent past. Several difficult problems have been attacked in this way:The Hamiltonian path [4, 5, 9, 7],The Travelling salesman problem [1, 9, 7],the subset sum [5, 3, 9, 7], 3-SAT [6], cryptography [10], factorization [2, 8] and some other NP-complete problems [5, 9].

Here we describe an optical implementation for the Eratosthenes sieve. The device is based on 2 parallel mirrors between which different beams of light are reflected. One of the mirrors also contains sensors which detect if the light has hit a certain region. If we send a beam at a certain angle, it will be reflected between those 2 mirrors multiple times. This process simulates the marking process of the Eratosthere Sieve. If we send multiple beams at different angles we can parallelize the marking process, thus providing a speedup compared to the sequential version of the algorithm.

The paper is structured as follows: Section 2 contains a short description of the problem and of the Eratosthenes sieve strategy. In the next section 3 we describe the proposed optical implementation. In section 4 we show the straights and weaknesses of the system. The final section 5 concludes our paper.

Received by the editors: February 14, 2011.

2010 *Mathematics Subject Classification.* 82D99, 11A41.

1998 *CR Categories and Descriptors.* F.2.1 [**Theory of Computation**]: Number-theoretic computations .

Key words and phrases. sieve of Eratosthenese, Optical computing.

2. Problem Description

A prime number is a number that can be divided only by 1 and itself. A basic algorithm for determining that a number is prime consists in trying all the numbers from 2 up to the square root of that number and checking if one of them divides it. If none is found we can say that the number is prime.

Sieve of Eratosthenes is a prime number strategy looking to find all prime numbers up to a number. The algorithm is quite simple and it looks like this: Having an array of natural numbers up to a limit, start with the number 2 and remove all its multiples(excluding it); after this step go to next unremoved number and remove all its multiples excluding itself; repeat this step until no next unremoved numbers exist. At the end, all the leftover (unremoved) numbers are prime.

The main difference between the proposed optical algorithm and Eratosthenes sieve is that the optical one will do the check for only one number. This limitation will be explained in section 3.4

3. The optical implementation

In this section we propose a solution for implementing the Eratosthene Sieve in an optical manner. We design a device in which we send beams of light which touch some regions at a regular distance.

The optical implementation is a combination between the sieve of Eratosthenes and the trial division. Trial division tests if an integer n can be divided by any integer greater than one but less than n. Having an array of natural numbers up to a limit, start with number 2 and remove all its multiples; after this go to next number and remove all its multiples including itself and repeat this step until the rounded square of the tested number. Eliminating a number that was not previously hit is a weakness imposed by the way the light is send.

Solving this problem on conventional computers requires an array. Each cell from that array corresponds to a natural number. At the beginning of the algorithm all cells are **true**. A loop iterates from 2 to the square root of the input number. All the cells that have an index which is a multiple of 2 are marked with **false**. After that, the loop selects the next cell in increasing order and marks all its multiples as **false**. At the end if the input number is not marked as **false** it means it is a prime number.

The pseudo code of the sieve of Eratosthenes used for the optical implementation looks like this:

v := an array of n elements(all **false**);
for i **from** 2 **to** $\lfloor\sqrt{n}\rfloor$
    if v[i] is **false**
        $k := i * 2$;

```
        while k < n do
            v[k]:= true;
            k := k+i;
        endwhile
    endif
endfor
```

at the end all the possitions that are false are prime numbers;

3.1. Required physical components. Basically we need the following components:

- 2 parallel mirrors,
- an array of sensors (photodiodes) placed on one of the mirrors,
- a prism,
- a source of light

In what follows we give details about each component.

3.2. Mirrors and sensors. The implementation is based on 2 parallels mirrors. One mirror is used just for reflections. The second mirror also reflects the light, but it also has a cell (sensor) which is capable of detecting if a beam of light hit it.

Each cell is placed at a regular interval. The distance between 2 consecutive cells is always the same. A cell has a simple function: it acts as a reflective mirror.The last cell is the number to be tested, and needs to be a photodiode in order to record if has been touched or not by light. The cell index is represented by a number. A number of cells equal to the input number to be tested is needed.

3.3. Angles of reflection. When a beam of light is sent under a known angle it will be reflected between the mirrors, touching them at a regular interval. The intervals where the mirror will be hit may be computed based on the distance between the mirrors and the angle at which the beam is sent.

An angle which will generate a step of n cells must be computed using the formula:

$$\sin x = \frac{d}{\sqrt{(\frac{s}{2})^2 + d^2}} \tag{1}$$

where:
x- is the angle at which the beam must be sent
d- is the distance between mirrors

s- is the step(represents distance). Note that s it is computed based on the distance between cells and the frequency for the current beam that touches the mirror. It may be computed with the formula:

s = distance between cells * frequency to hit

Example: if the distance between cells is 2 μ and a beam that hits all divisors of 5 is needed, s will be: 5*2 = 10μ

d and s must use the same unit of measurement

3.4. **Sending the light.** We have, inside the device, multiple rays at different angles. All angles encoding numbers from 2 to $sqrt(n)$ must be generated. Obtaining light at multiple angles can be done by using a prism. Prisms must also deflect a beam of light by a given angle.

Also we have discussed earlier (see the begining of section 3) about a limitation imposed by the way the light is sent, that has as a consequence the usage of the algorithm for a single number. This happens because we send all the light beams from the position 0.

If the avoidance of uncut numbers is desired, this will have unwanted consequences. One of them would be that the light will not be sent in a parallel manner because avoiding a cell should be done based on the previous results. Another limitation would be the need of physical movement of the light emitter.

3.5. **How the device works.** All beams are sent to the device through position 0. After that, the beams will traverse the device and will mark several positions (by hitting the corresponding sensors). At the end of the operation if the number we are interested in is not touched by any light, it is prime.

A basic example is shown in Figure 3.5. The example means to test the input number: 17. In order to test this number we have to send three beams of light. The numbers that must be tested starts from 2 and goes untill the truncated value from the square root of 17 which is 4. So we have to test the division by 2, 3 and 4.

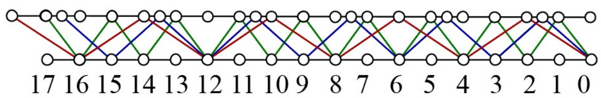

FIGURE 1. Optical implementation for the Eratosthenes Sieve. Bottom mirror contains optical detectors. 3 different beams have been sent (in the same time) from point 0 and we have depicted their path. Numbers which are multiples of 2 (green), 3 (blue) and 4 (red) are marked here.

Interference between two beams will not happen if we send short enough pulses. If we send two short beams from the same point, under different angles, the one that is reflected more rarely will travel faster because it has less distance to cover between mirrors until it gets out.

4. Strengths and weaknesses

The most important benefit of the algorithm is the speed generated by its massive parallelism. It allows us to send all the beams at the same time.

On a conventional computer, for the best parallelized algorithm, the speed up is in the best case equal to number of cores. On this optical system the concurrent speed-up is equal to the numbers of light beams. And there is the possibility to send all the beams at the same time. This means that the gained speed is equal to the square root of the number to be tested.

There are a number of weak points. The most important is that only small problems can be solved. That is because of some technical chalenges: reflection, absorbtion, etc.

5. Conclusions

We have exemplified how the very popular Eratosthene sieve's algorithm can be easily implemented with optics.

Future work will be focused on making a harware implemention and on increasing the parallelism of the system.

References

[1] Haist, T., Osten, W.: An Optical Solution For The Traveling Salesman Problem. Opt. Express, Vol. 15, 10473-10482 (2007)

[2] Kouichi Nitta, Nobuto Katsuta, Osamu Matoba: Improvement of a System for Prime Factorization Based on Optical Interferometer. LNCS 5882 Springer, pp. 124-129, 2009

[3] Md. Raqibul Hasan, M. Sohel Rahman, Computing a Solution for the Subset Sum Problem with a Light Based Device. LNCS 5882 Springer, pp. 70-76, 2009

[4] Oltean, M.: A light-based device for solving the Hamiltonian path problem. Unconventional Computing, Calude C. (et al.) (Eds), LNCS 4135, Springer-Verlag, 217-227 (2006)

[5] Oltean, M., Muntean, O.: Solving NP-Complete Problems with Delayed Signals: An Overview of Current Research Directions, in proceedings of 1st international workshop on Optical SuperComputing, LNCS 5172, pp. 115-128, Springer-Verlag, 2008

[6] Sama Goliaei, Saeed Jalili: An Optical Wavelength-Based Solution to the 3-SAT Problem. LNCS 5882 Springer, pp. 77-85. 2009

[7] Shaked, NT., Messika, S., Dolev, S., and Rosen, J.: Optical solution for bounded NP-complete problems, Applied Optics, Vol. 46, 711-724 (2007)

[8] Shamir, A., Tromer, E.: Factoring Large Numbers with the TWIRL Device, LNCS 2729 Springer-Verlag, pp1-26. 2003

[9] Shlomi Dolev, Hen Fitoussi: The Traveling Beams Optical Solutions for Bounded NP-Complete Problems. FUN 2007: 120-134

[10] Tobias Haist, Wolfgang Osten: Proposal for Secure Key Distribution Using Classical Optics. LNCS 5882 Springer, pp. 99-101, 2009

Department of Computer Science, Faculty of Mathematics and Computer Science, Babeş-Bolyai University, Kogălniceanu 1 Cluj-Napoca, 400084, Romania

E-mail address: liviu@cs.ubbcluj.ro