Light-based string matching

Mihai Oltean

Department of Computer Science
Faculty of Mathematics and Computer Science,
Babeş-Bolyai University, Kogălniceanu 1
Cluj-Napoca, 400084, Romania.
moltean@cs.ubbcluj.ro

Abstract. String matching is a very important problem in computer science. The problem consists in finding all the occurrences of a pattern P of length m in a text T of length n. We describe a special device which can do string matching by performing $n-m+1$ text-to-pattern comparisons. The proposed device uses light and optical filters for performing computations. Two physical implementations are proposed. One of them uses colored glass and the other one uses polarizing filters. The strengths and the weaknesses of each method are deeply discussed.

1 Introduction

String matching is the problem of finding all the occurrences of a pattern $P[1 \ldots m]$ in a text $T[1 \ldots n]$ [3, 8, 19]. The problem is very practical in its nature: it occurs in many real-worlds applications such as web search engines, linguistics, bioinformatics etc. This is the reason why algorithms should be efficient even if the speed and capacity of storage of computers increase regularly.

We are focused on the offline matching which is the most common variant of the problem. In this variant the text and the pattern are known in advance and they can be accessed in an unrestricted manner.

Theoretically, at least $n-m+1$ comparisons should be performed for solving this problem [3]. However, in practice, there is currently no algorithm which performs less than n comparisons on a conventional computer.

We show how to solve this problem by performing only $n-m+1$ comparisons. For achieving this performance we propose two special devices which use the inherent properties of light for performing computations. Both rely on optical filters [4, 18] for comparing characters. One of them uses colored glass and the other one uses polarizing filters.

An optical filter is a device which selectively transmits light having certain properties, while blocking the remainder.

In the case of colored glass only light of the same color with the glass will pass through. If we put two pieces of glass, of different colors, one in the front of the other, no light will pass the second glass [18, 29].

A similar property holds for the polarizing filters as well. If two filters are aligned they will let the light to pass through. If the filters are perpendicular one on the other no light will pass through [10].

The operations described above can be used as basic building blocks for the comparison of characters and strings of characters. We use the massive parallelism of light in order to perform these comparisons in parallel.

The paper is structured as follows: section 2 contains a brief overview of the complexity of string matching algorithms. A review of some interesting light-based computing devices is presented in section 3. The knowledge required for practical implementation of the proposed devices is given in section 4. We start the description of our idea in section 5 where we explain how to compare 2 characters using our device. We improve our idea in sections 6 and 7 where we show how to deal with equal-size string and with strings of different lengths. The drawbacks and some solutions for the physical implementation are discussed in section 8. Conclusions and further work directions are given in section 9.

2 String matching: algorithms and complexity

Various algorithms for the string matching problem have been proposed. The basic idea is the following: the pattern is initially aligned with the left end of the text. Repeatedly, an attempt to match the pattern against the text is made. The pattern is shifted to the right when a mismatch is found or when the pattern is fully matched. In almost all the algorithms the goal is to shift the pattern with the maximal number of positions without missing any possible matches.

Perhaps the best known linear-time algorithms for string matching are the Knuth-Morris-Pratt [19] and Boyer-Moore [3] algorithms [5]. We refer to these as the KMP and BM algorithms, respectively. The KMP algorithm makes at most $2n - m + 1$ comparisons and this bound is tight. The exact complexity of the BM algorithm was an open question until recently. It was shown in [19] that the BM algorithm makes at most $6n$ comparisons if the pattern does not occur in the text. In [17] the bound was reduced to $4n$ and this result was later refined in [7] where a bound of $3n - \Omega(n/m)$ comparisons was proven. In [7] a simple variant of the KMP algorithm which makes at most $3n/2$ comparisons is given. Another paper [2] gave a variant of the BM algorithm which makes at most $2n - m + 1$ comparisons. In [9] it was shown that, by remembering the most recently matched portion, we can reduce the upper bound of BM from $3n$ to $2n$ comparisons.

A string-matching algorithm which makes at most $4n/3$ comparisons was proposed in [14]. Galil and Giancarlo [13] gave a lower bound of $n(1 + 1/(2m))$ comparisons. This bound shows that there is yet no algorithm which performs less than n comparisons for finding all occurrences of the pattern inside the text.

3 Computing with light-based devices

All major computational devices are nowadays using electric power in order to perform useful computations. Another idea is to use light instead of electrical power. It is expected that optical computing could mark a breakthrough in

computer architecture and that could improve the speed of data input and output by several orders of magnitude [12].

Many theoretical and practical light-based devices have been proposed for dealing with various problems. Optical computation has some advantages, one of them being the fact that it can perform some operations faster than conventional devices. An example is the n-point discrete Fourier transform computation which can be performed in only one unit time [16, 26].

The quest for the light-based computer was started in 1929 by G. Tauschek who obtained a patent on Optical Character Recognition (OCR) in Germany. Next step was made by Handel who obtained a patent on OCR in USA in 1933 (U.S. Patent 1,915,993). Those devices were mechanical and used templates for matching the characters. A photodetector was placed so that when the template and the character to be recognized were lined up for an exact match, and a light was directed towards it, no light would reach the photodetector [32].

An important practical step was made by Intel researchers (see Figure 1 (a)) who have developed the first continuous wave all-silicon laser using a physical property called the Raman Effect [11, 25, 27, 28]. The device could lead to such practical applications as optical amplifiers, lasers, wavelength converters, and new kinds of lossless optical devices.

Another solution comes from Lenslet [20] which has created a very fast processor for vector-matrix multiplications (see Figure 1 (b)). This processor can perform up to 8000 Giga Multiple-Accumulate instructions per second. The Lenslet technology has already been applied to data analysis using $k-$mean algorithm [21] and video compression.

A recent paper [30] introduces the idea of sorting by using some properties of light. The method called Rainbow Sort is an unconventional method of sorting, which is based on the physical concepts of refraction and dispersion. It is inspired from the observation that the light traversing a prism is sorted by wavelength (see Figure 1 (c)). For implementing the Rainbow Sort one needs to perform the following steps:

- the encoding of multiple wavelengths (representing the numbers to be sorted) into a light ray,
- the sending of the ray through a prism which will split the ray into n monochromatic rays that are sorted by wavelength,
- the receiving of the incoming rays by a detector on the output side.

A stable version of the Rainbow Sort is proposed in [22].

Naughton (et al.) proposed and investigated [23, 33] a model called the continuous space machine which operates in discrete timesteps over a number of two-dimensional complex-valued images of constant size and arbitrary spatial resolution. The (constant time) operations on images include the Fourier transformation, multiplication, addition, thresholding, copying and scaling.

In [24] a special computational device which uses light rays for solving the Hamiltonian path problem on a directed graph was proposed. The device has a graph-like representation and the light is traversing it following the routes given

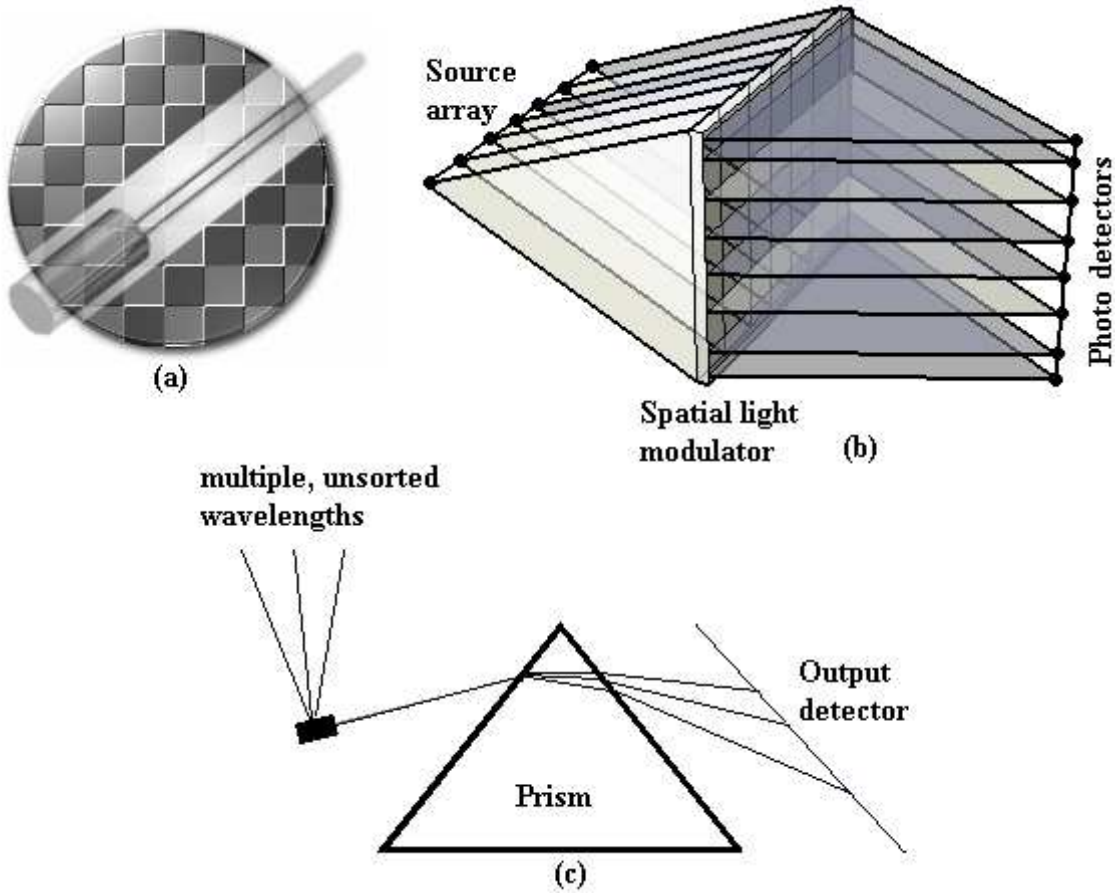

Fig. 1. (a) Intel concept: a continuous wave all-silicon laser. (b) A sketch of the Lenslet device used for performing vector-matrix multiplications. (c) Schematic view of the Rainbow Sort.

by the connections between nodes. The rays in each node are uniquely marked so that they can be easily identified. At the destination node the device searches only particular rays that have passed only once through each node. The device could solve small and medium instances of the problem in reasonable time.

4 Basic components of the proposed device

4.1 Colored glass

It is widely known that colored glass acts like a wavelength filter. For instance, a red glass will allow light having a wavelength of about 650 nm to pass through while being absorbed in the lowest degree. The other wavelengths will be absorbed with a greater efficiency.

In our string matching problem we deal with arrays of characters over a finite alphabet A. To each character in the alphabet we associate a piece of glass of a given color. For instance, if the alphabet is $A = \{a, b, c\}$ we will have glasses of 3 different colors (for instance: red, yellow and blue). The colors used to represent each character are not important. What is important is the fact that two different characters must have pieces of glass of two different colors associated with them.

4.2 Polarizing filters

Another idea for our device is to use polarizing filters instead of variously colored pieces of glass.

The idea is fairly similar: if two polarizing filters are aligned (the same angle) they will let the light pass through. If the filters are not aligned (they have different angles), the second filter will absorb part of the light. In the worst case, if the filters are perpendicular to each other no light will pass through.

For our purpose we have to uniquely represent each character from the considered alphabet A. We assign to each character from the given alphabet A a polarizing filter which is rotated by a certain angle (between 0^o and 90^o). For instance if the alphabet contains 3 characters we may use filters which are rotated by 0^o, 45^o and 90^o.

5 Testing the equality of 2 characters

We have 2 characters and we want to find if they are the same. This is an extremely simple operation for a computer, but in our proposal we have to construct a special device for it.

5.1 Comparing using colored glass

We take two pieces of glass having the colors associated to those 2 characters involved in the comparison. We put one glass in the front of the other and we direct a ray of light towards one of them (see Figure 2). If the ray will pass

through both pieces of glass it means that they have the same color (thus they represent the same character). If the light does not pass it means that the pieces of glass have different colors. We test the fact that the light has passed through both pieces of glass by using a photodiode.

In this way we have reduced the problem of testing if 2 characters are equal to the problem of finding if two pieces of glass have the same color.

Remark

The entire device must be isolated from the external sources of light. Also the light which is sent to the colored glasses must not be able to reach the photodiode by any other means. This can be avoided if we put the entire device in a box having the same size as the pieces of glass.

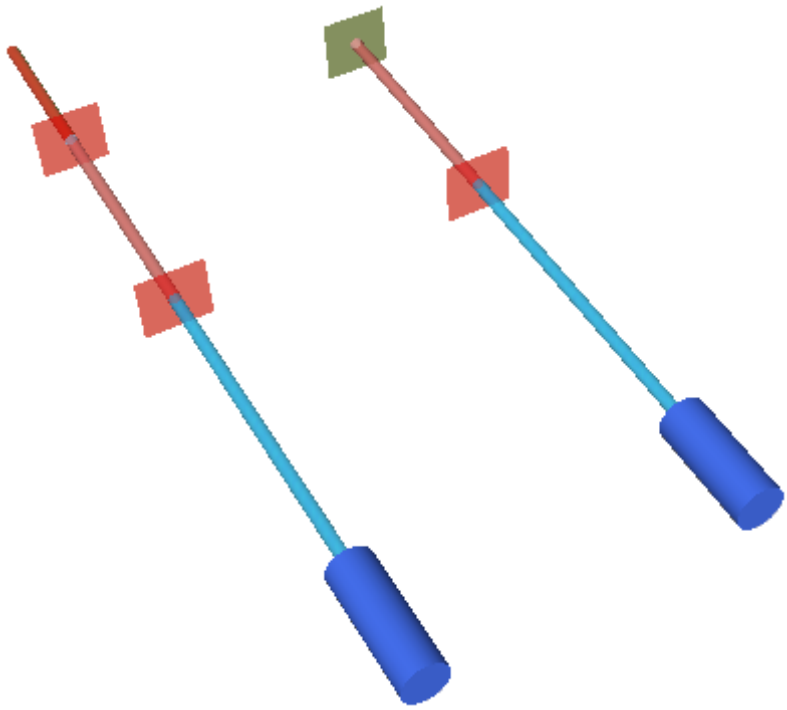

Fig. 2. Special devices used for testing the equality of 2 characters. If the pieces of glass have the same color (actually representing the same character) the light will pass through both of them (see the left side of the picture). If the pieces of glass have different colors the light will not pass through the second one (see the right side of the picture).

5.2 Comparing by using polarizing filters

We take two filters which are rotated to some degree which are associated to those 2 characters involved in the comparison. We put one filter in the front of the other and we direct a ray of light towards one of them (see Figure 3).

Two possibilities arise:

- the light having passed the first filter has the same intensity as one having passed the second filter (in this case the filters have the same alignment),
- the light having passed the second filter has a lower intensity than the one having passed the first filter (in this case the filters have different angles).

We test the light intensity at the exit of the second filter by using a photodiode.

In this way we have reduced the problem of testing if 2 characters are equal to the problem of finding if two polarizing filters are rotated by the same angle.

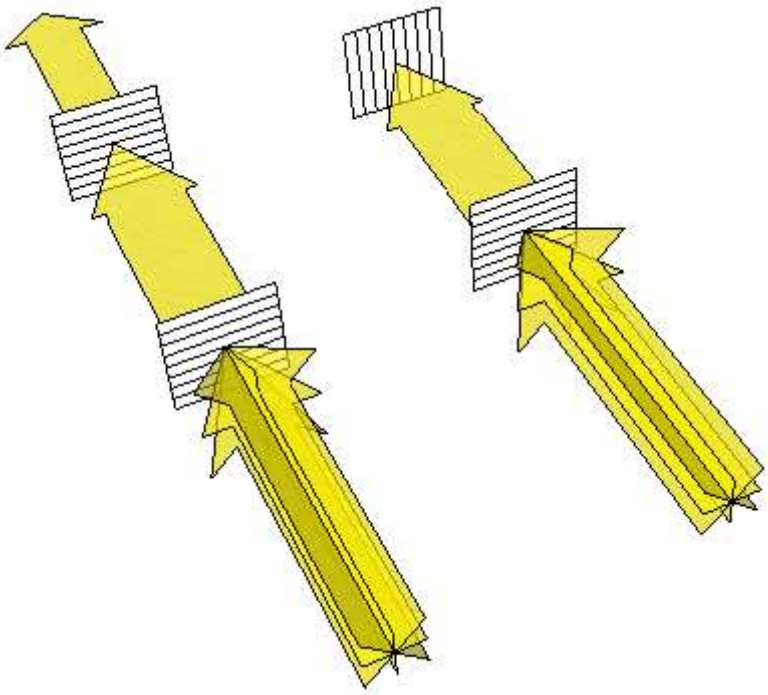

Fig. 3. Special devices used for comparing 2 characters. Unpolarized light is sent to filters. If the filters are aligned the light will pass through both of them (see the left side of the picture). If the filters are perpendicular one on the other, no light will pass through (see the right side of the picture). The intensity of light which passes the second filter depends on the angle between filters (which is restricted to the interval $[0^o, 90^o]$: if the angle is increased, the intensity will decrease.

6 Testing the equality of 2 equal-size strings

We have 2 strings of characters having the same length (denoted by n) and we want to find if they are identical. By using a classic computer we need to do at most n comparisons. We will show how to do this operation by performing one comparison only.

6.1 Comparing by using colored glass

Each string will be represented by a sequence of pieces of glass having the colors associated to the characters in alphabet. We put these sequences face to face and we send some light toward one of them (see Figure 4). For this operation we use an electric bulb placed at such a distance from the panels such that the rays which are passing through pieces of glass are almost parallel. Some of the rays will pass through both pieces of glass and some will not pass at all.

We focus the rays which have passed by using a convex lens. In the focal point we place a photodiode. By measuring the electric power we can tell if the corresponding strings are equal or not. Thus we have to perform a single comparison.

6.2 Comparing by using polarizing filters

The pattern and the text will be represented by two sequences of polarizing filters having the angles which are associated to the characters in the considered alphabet. We put these sequences face to face and we send some light toward one of them (see Figure 5). The intensity of the light which passes the second sequence of filters depends on the angle of the alignment of the filters. Basically speaking the intensity of light which passes through is maximal if all the pairs of filters (one from the first sequence and the corresponding one from the second sequence) have the same angle.

Then we apply the same procedure as in the case of colored glass (see section 6.1): we focus the rays which have passed the second sequence of filters by using a convex lens. In the focal point we place a photodiode. By measuring the electric power we can tell if the corresponding strings are equal or not. Basically speaking the strings are equal if the light intensity is maximal.

Thus, we perform only one comparison again.

7 General string matching

Now we have to solve the general case where we have a pattern P of length m and a text T of length n. For finding all the occurrences of P in T we consider every possible initial position of an occurrence of pattern in text. We have $n - m + 1$ such positions. For each initial position k we compare P with $T_{k...k+m-1}$. This comparison can be made in one step by applying the procedure described in section 6.

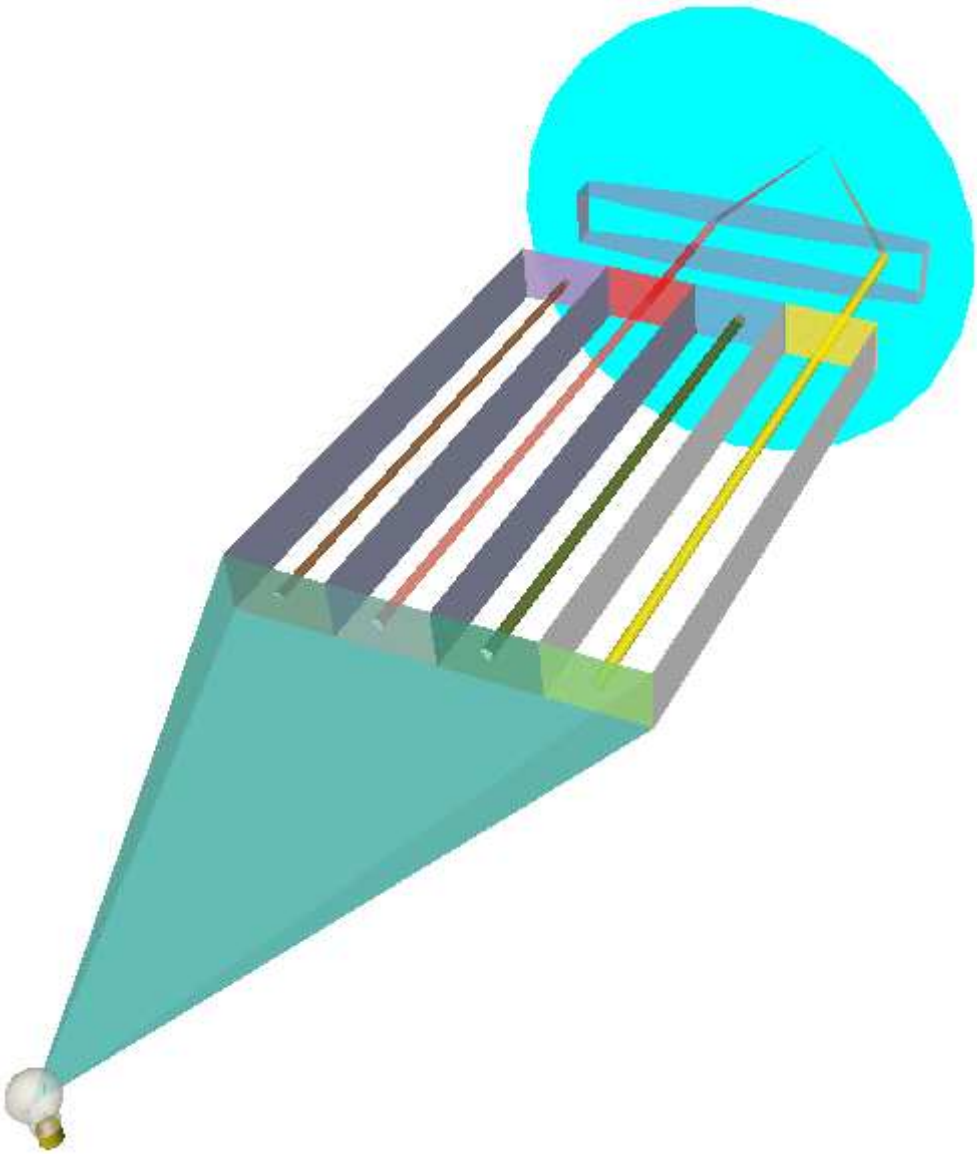

Fig. 4. The special device used for testing if 2 sequences of 4 characters are equal. For this purpose we need an electric bulb, 8 transparent pieces of glass having the colors corresponding to the characters, a lens and a photodiode. This picture shows how the light behaves when it passes through pieces of glass. The light changes its color when it passes through the first sequence of glasses. In only 2 cases (out of 4) the light is able to pass both pieces of glass (because in the other 2 cases the pieces of glass have different colors). A lens is used in order to focus the rays. In the focal point of the lens we have a photodiode.

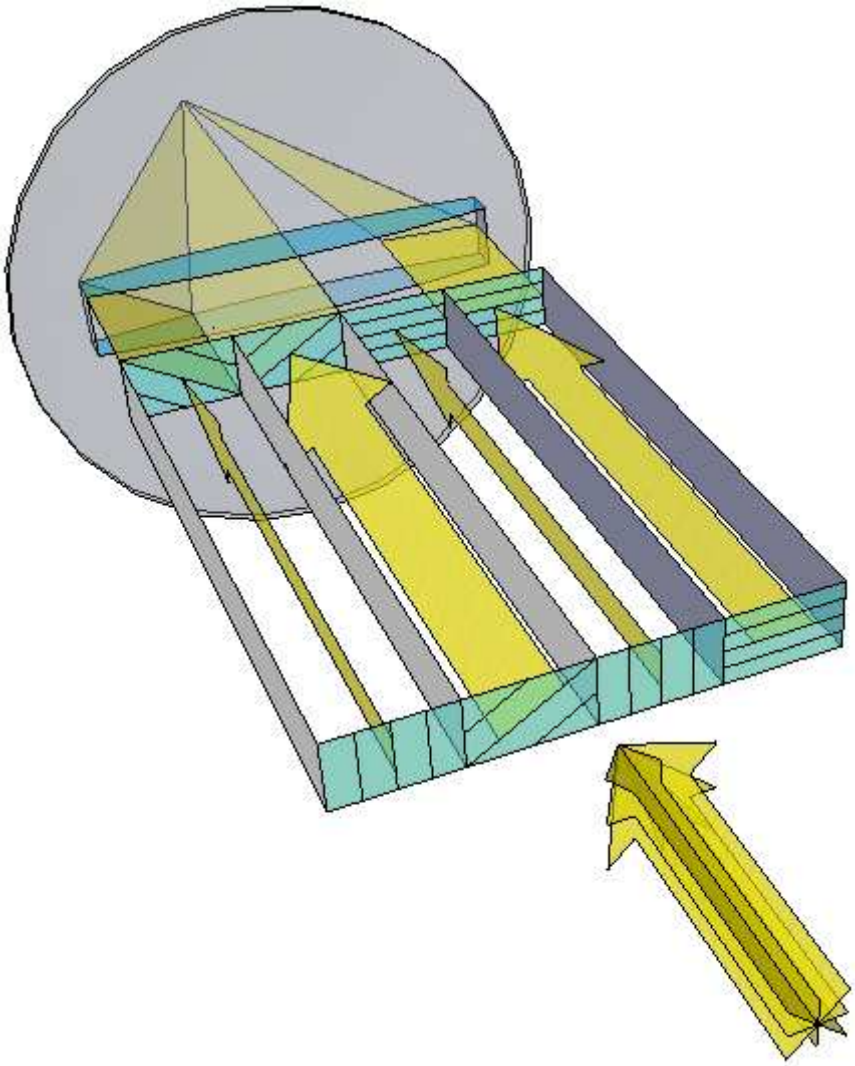

Fig. 5. Testing the equality of 2 strings of length 4. Eight polarizing filters, a lens, a source of light and a photodiode are used for this purpose. If the intensity of light, which is focalized by the lens, is at maximal intensity it means that all pairs of filters have the same rotation angle (basically speaking the strings are equal). Otherwise the strings are not equal.

Thus the total number of comparisons is $n - m + 1$.

For building the sequences of filters we need to perform $n+m$ steps. This operation is called preprocessing. Note that other algorithms are using preprocessing for speeding up the comparison phase [3, 19].

8 Difficulties in physical implementation and proposed solutions

The proposed methods have a reduced number of comparisons due to the special properties of light and optical filters, but there are some difficulties which are encountered during the physical implementation.

8.1 General difficulties

The device currently deals with fixed-length patterns and texts. Changing the number of characters in these strings is difficult for our model. In this case we have to build another 2 sequences of colored glasses or polarizing filters which is a time expensive operation even if it is performed in linear time.

However, there is one case in which we can reuse the already built sequence of polarizing filters: if the lengths of the pattern and text are not changed we can change the polarization plan (actual the character encoded in that position) by using the so called Faraday effect (or rotation) [18]. This effect is actually an interaction between light and magnetic field and it consists in the rotation of the polarization plan. The rotation is proportional with the intensity of magnetic field. Using this effect we can change an existing sequence of characters into another one. However, a new problem appears in this case: if the devices generating the magnetic field are too close they can interfere, thus, the polarization plan can change into an unwanted (and possible unpredicted) way.

Another solution is to first build 2 very large sequences of polarizing filters. Each time when a new string matching problem has to be solved we will set the polarization plan for the first n (respectively m) positions in the constructed sequences by using the Faraday effect. This solution is used by other computing devices. For instance, Lenslet's processor (see section 3) does vector-matrix multiplications for up to 256 elements in the vector and up to 256x256 elements in the matrix. The advantage of Lenslet's device is speed, but it also suffers from the same drawback: the number of elements in input cannot be increased.

Increasing the number of characters in the pattern requires an increased diameter for the lens. This is another difficulty for our system.

Shifting required when we have to compare patterns and texts of different sizes (see section 7) is another time consuming operation because it requires a motor (or another mechanical device) which moves the pattern one position each time.

8.2 Difficulties for colored pieces of glass

One problem is related to the capacity of colored glass to absorb all wavelengths with one exception. Unfortunately this operation cannot be done in a flawless way: a wavelength is not completely absorbed even if it is different from the color of the piece of glass. There is a degree to which wavelengths are absorbed by the colored glass. For avoiding this problem we have to use pieces of glass which have distant colors on the wavelength filter.

Another question is how the photodiode will react when the light rays of different wavelengths hit its surface.

The answer is very simple but it poses more difficulties to the methods: the reaction depends on the wavelengths and on the sensitivity of the photodiode to those wavelengths.

Photodiodes are made from semiconductors such as silicon, germanium, and InGaAs (Indium Gallium Arsenide). Each of these semiconductors has a bandgap energy. A photodiode will typically work for any wavelength where the corresponding photon energy is greater than bandgap energy. So, if two wavelengths are not separated much, the same photodiode will generate roughly the same amount of power.

If, however, the wavelengths are significantly separated such as red ($\lambda = 650nm$) and green ($\lambda = 514nm$), then the difference will become important.

We also have to take into account the fact that a photodiode will work for photon energies much larger than bandgap energy. For instance, a silicon detector will detect both wavelengths (green and red), but germanium and InGaAs will not detect them.

Next, we have to consider the responsiveness of the detector. Responsiveness measures how many amperes of power will be generated per watt of incident light. Unfortunately, the responsiveness is wavelength dependent. Therefore, given equal incident powers of red and green light, a silicon photodiode will generate different powers [18, 29].

Taking into account the previously discussed aspects one can easily see that is difficult to capture light having different wavelengths on the same photodiode. One idea is to use m photodiodes (one for each pair of filters), each of them acting as a switch for other m sources of light. Each of these small devices (photodiode + source of light) will be placed just before the light reaches the lens (see Figure 4). If the light which is passing both filters has intensity over a given threshold (which ensures that the filters have the same color) it will activate the new source of light which is generating some white light. These new rays will be captured by the terminal photodiode in the focal point of the lens.

8.3 Difficulties for polarizing filters

There are also few difficulties related to the polarizing filters. The difficulties appear when the filters have very close rotation angles. In this case it might be difficult to find out if the filters are aligned (the same rotation angle) or not. However this problem could be avoided, by using a high-accuracy photodiode.

The photodiodes are not sensitive to the polarization angle, thus there is no problem related to this issue whatsoever.

9 Conclusions and further work

In this paper we have proposed two devices which use the inherent properties of light in order to solve a well-known computer science problem. Two kinds of optical filters have been used for our purpose: colored glass and polarizing filters. We have made an in-depth analysis of the strengths and of the weaknesses of each method. At first sight we can infer that polarizing filters are more stable than colored glass for the string matching problem.

The physical implementation of the proposed devices might be time consuming, so these methods might not bring such a great benefit unless we find some real-world cases where there are no other options for implementation but the ones we have proposed. However, the greatest benefit is that we have shown that string matching can be efficiently done by using the massive parallelism of the light.

Further work directions will be focused on:

- Implementing the proposed device,
- Finding better ways to improve the robustness of comparisons, especially in the case of colored glass,
- Finding better ways to implement the shifting,
- Finding a way to deal with variable length strings,
- Finding other problems which can be solved using similar principles.

References

1. Agrawal, G.P., Fiber-optic communication systems, Wiley-Interscience; 3^{rd} edition, 2002
2. Apostolico, A. and Giancarlo, R., The Boyer-Moore-Galil string searching strategies revisited, SIAM J. Computing, 15, pp. 98-105, 1986
3. Boyer, R. and Moore, S., A fast string matching algorithm, Comm. Assoc. Comput. Mach., 20, pp. 762-772, 1977
4. Born, M. and Wolf, E., Principles of Optics, 7^{th} edition, Cambridge University, 1999
5. Cole, R., Hariharan, R., Paterson, M. and Zwick U., Tighter lower bounds on the exact complexity of string matching, SIAM Journal of Computing, Vol. 24, Issue 1, pp 30-45, 1995
6. Cole, R. and Hariharan, R., Tighter upper bounds on the exact complexity of string matching, SIAM Journal of Computing, Vol. 26, No. 3, pp. 803-856, 1997
7. Colussi, L., Correctness and efficiency of pattern matching algorithms, Inform. and Comput., 5, pp. 225-251, 1991
8. Cormen, T.H., Leiserson, C.E., Rivest R.R., Introduction to algorithms, MIT Press, 1990

9. Crochemore, M., Czumaj, A., Gasiniec, L., Jarominek, S., Lecroq, T., Plandowski, W. and Rytter, W., Speeding up two string-matching algorithms, Algorithmica, 5, pp. 247-267, 1994
10. Damask, J.N., Polarization Optics in Telecommunications, Springer, 2004
11. Faist, J., Optoelectronics: silicon shines on, Nature, Vol. 433, 691-692, 2005
12. Feitelson, D.G., Optical Computing: A Survey for Computer Scientists, MIT Press, 1988
13. Galil, Z. and Giancarlo, R., On the exact complexity of string matching: Lower bounds, SIAM J. Comput., 6, pp. 1008-1020, 1991
14. Galil, Z. and Giancarlo, R., On the exact complexity of string matching: Upper bounds, SIAM J. Comput., 3, pp. 407-437, 1993
15. Garey, M.R., Johnson, D.S., Computers and intractability: A guide to NP-Completeness, Freeman & Co, San Francisco, CA, 1979
16. Goodman, J. W., Architectural development of optical data processing systems, Aust. J. Electr. Electron. Eng. 2, 139149, 1982
17. Guibas, L.J. and Odlyzko, A. M., A new proof of the linearity of the Boyer-Moore string searching algorithm, SIAM J. Comput., 9, pp. 672-682, 1980
18. Hecht, E., Optics, Addison Wesley, 4^{th} edition (2002)
19. Knuth, D.E., Morris (Jr) J.H., Pratt, V.R., Fast pattern matching in strings, SIAM Journal on Computing 6(1), 323-350, 1977
20. Lenslet website, www.lenslet.com
21. MacQueen, J., Some methods for classification and analysis of multivariate observations, In LeCam, L. M., Neyman, J., (eds.) Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. University of California press, Berkeley, 281-297, 1967
22. Murphy, N., Naughton, T. J., Woods, D., Henley, B., McDermott, K., Duffy, E., van der Burgt, P. J. M., and Woods N., Implementations of a model of physical sorting, From Utopian to Genuine Unconventional Computers workshop, Adamatzky, A., and Teuscher, C., (editors), Luniver Press, 79-100, 2006
23. Naughton T. J., A model of computation for Fourier optical processors, In Lessard, R. A. and Galstian, T., (editors), Optics in Computing, Proc. SPIE vol. 4089, 24-34, 2000
24. 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
25. Paniccia, M., Koehl, S.: The silicon solution, IEEE Spectrum, IEEE Press, October (2005)
26. Reif, J.H. and Tyagi, A., Efficient parallel algorithms for optical computing with the discrete Fourier transform primitive. Applied optics, Vol. 36(29), 7327-7340, 1997
27. Rong, H., Jones, R., Liu, A., Cohen, O., Hak, D., Fang, A. and Paniccia, M., A continuous-wave Raman silicon laser, Nature, Vol 433, 725-728, 2005
28. Rong, H., Liu, A., Jones, R., Cohen, O., Hak, D., Nicolaescu, R., Fang, A. and Paniccia, M., An all-silicon Raman laser, Nature, Vol. 433, (2005) 292-294
29. Serway, R.A. and Jewett, J.W., Physics for scientists and engineers, 6^{th} edition, Brooks/Cole, 2004
30. Schultes, D., Rainbow Sort: Sorting at the Speed of Light, Natural Computing, Springer, Vol 5 (1)., pp. 67-82, 2005
31. Zwick, U. and Paterson, M., Lower bounds for string matching in the sequential comparison model, manuscript, 1992.

32. Optical Character Recognition @ Wikipedia,
http : //en.wikipedia.org/wiki/Optical_character_recognition
33. Woods, D., and Naughton, T.J., An optical model of computation, Theoretical Computer Science, Vol. 334, Issues 1-3, pp. 227-258, 2005