Dariusz FREJLICHOWSKI, Piotr DZIURZAŃSKI
POLITECHNIKA SZCZECIŃSKA, WYDZIAŁ INFORMATYKI

System-level implementation of a partial pattern matching algorithm

Dr inż. Dariusz FREJLICHOWSKI

Ukończył studia na Wydziale Informatyki Politechniki Szczecińskiej w 2001 r. Obronił pracę doktorską na tym samym wydziale w 2005 r. Obecnie jest adiunktem w Instytucie Grafiki Komputerowej i Systemów Multimedialnych WI PS. Jego zainteresowania naukowe to algorytmy przetwarzania i rozpoznawania obrazów, szczególnie z użyciem deskryptorów kształtów dla obrazów binarnych.

e-mail: dfrejlichowski@wi.ps.pl

Dr inż. Piotr DZIURZAŃSKI

Ukończył studia na Wydziale Informatyki Politechniki Szczecińskiej w 2000 r. Na tym samym wydziale obronił pracę doktorską w 2003 r. Obecnie pracuje na stanowisku adiunkta w Instytucie Architektury Komputerów i Telekomunikacji WI PS. Jest członkiem organizacji IEEE i ACM. Jego zainteresowania to kosynteza sprzętowo-programowa, synteza wysokiego poziomu, synteza logiczna i formalna weryfikacja.

e-mail: pdziurzaski@wi.ps.pl

Abstract

In this paper, we describe Partial Point Matching Algorithm (PPMA) for binary images matching and provide some information about a developed SystemC system-level hardware implementation model. We present some data obtained from a cycle accurate simulator taking into account computation and routing delays of a target FPGA chip and compare it with its software counterparts.

Keywords: pattern recognition, shape descriptors, SystemC, system-level design.

Implementacja algorytmu porównywania binarnych obrazów PPMA na poziomie systemowym

Streszczenie

W artykule opisano algorytm częściowego dopasowywania z użyciem punktów do porównywania obrazów binarnych i przedstawiono opracowany model na poziomie systemowym w języku SystemC. Zaprezentowano dane otrzymane z symulatora pracującego na poziomie dokładności co do cyklu zegarowego, biorącego pod uwagę opóźnienia bramek i rutingu, oraz porównano go z jego programowym odpowiednikiem.

Słowa kluczowe: rozpoznawanie obrazów, deskryptory kształtu, SystemC, projektowanie na poziomie systemowym.

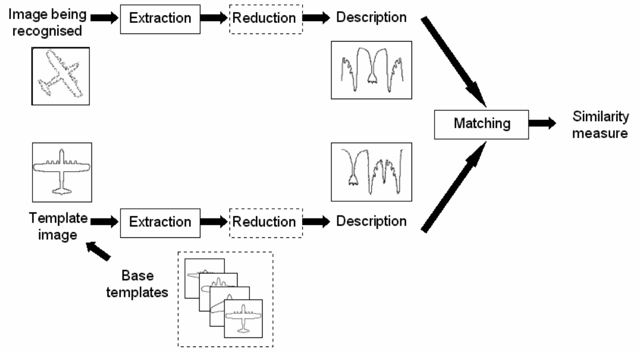
1. Introduction and motivation

A very popular and effective scheme of image recognition is based on template matching and comprised of two independent stages. In the first stage, the image to be recognized is extracted and appropriately represented. In the second stage, a comparison between that representation and the patterns stored in a data base is performed. Between these stages, it is possible to introduce a process for reducing the information (Fig. 1).

It may be assumed that the image or the object in it is sufficient for recognition. However, this approach is rarely used, as almost in every real case there are some problems hindering the recognition, such as an influence of light, a placement of the object in the picture, etc.

Consequently, it is usually assumed that the representation obtained as a result of the extraction process is to be independent on these problems as much as possible. Each representation is usually based on a particular feature. The most popular features are color, texture, movement, luminance, and shape among others.

If we use a shape to represent an object, we have to take into account various conditions, such as a placement and orientation of the object into an image, its different size, deformation of its shape, e.g. a noise at its edges or a lack of a part of it. The object representation based on a shape, so called shape descriptor, is required to be independent of the maximal possible number of the above mentioned problems [4].



Rys.1. Typowy schemat rozpoznawania obrazów opartego na porównywaniu wzorców
Fig. 1. Typical scheme of image recognition based on template matching

The primary advantage of the approach based on shape descriptors is the fact that the size of the data base is limited. Instead of increasing its size with adding new deformed version of the object, we try to find such its representation that is maximally independent on these deformations.

Among numerous shape descriptors, the method using a transformation to the polar coordinate system can be very promising [5]. In literature, a number of algorithms based on this transformation are described, such as mUNL [2]. It is invariant to translating an object in image due to performing a transformation according to an assumed point. A rotation in the Cartesian coordinate system becomes a cyclic shift in the polar system, so a dedicated technique for comparison with a template image is used, which is independent on this problem. Utilizing a matrix representation leads to an independence on a starting point choice and the direction of a tracing the contour, whereas applying interpolation results in decreasing an influence of local shortage of points at the object edge on the matching results. Being a different from classic technique for determining the center of a transformation, mUNL is also quite robust to occlusion (i.e., some pixels excess or shortage in the contour).

Nevertheless, there still remain two problems, namely, the rotation-translation and the noise. Both of these problems are solved with Partial Point Matching Algorithm (PPMA) as the method for matching. This approach has been successfully used not only in recognition of various images (e.g., trademarks, license plates, safety tags, planes silhouettes, banknotes, etc.), but also in determining similarities between curves in stock exchange charts.

Despite having a number of assets, the chief disadvantage of PPMA is its computational complexity. According to this algorithm, the image being recognized is shifted cyclically the number of times equal to the number of rows in the image. This is why we have tried to implement it in hardware so as to perform the computation concurrently utilizing both the coarse- and fine-grain parallelism, as described in the sequel of this paper.

2. PPMA algorithm

The PPMA algorithm works at the level of separate pixels, which aims at maximizing the independence on the occlusion problem. On input we provide a square matrix with binary data. In case of image recognition, it is representation of the contour object obtained with, e.g., the mUNL transformation. If a pixel belongs to the object, we assign 0 into the appropriate matrix element, and 1 otherwise, i.e., for white background pixels.

The first step in determining the similarities between two representations is expanding the number of points belonging to the descriptor of the image representation from the data base. It inserts value 1 in the descriptor matrix for every background pixel in both vertical directions. The number of inserted elements is equal to the noise toleration level chosen a priori. For example, if this level is chosen to be equal to 2, for every non-zero pixel we insert 4 new elements in its neighborhood (2 above and 2 below).

In the second stage of the described algorithm, the logical conjunction (AND) operation between two expanded objects, the template image and the image to be recognized, is performed. Both these matrices are of the same size. The number of obtained 1s (i.e., the black elements) determines the degree of similarity between these objects. To take into consideration a rotation in the image plain, we have to perform a cyclic shift of the image being recognized by 1 pixel in right and repeat this operation. If we assume the size of the input data of 128x128 pixel, 127 shifting operations are performed. Then, we compute 128 different values of similarity degree, choose the largest and assume it as the final degree of similarity between the image being recognized and the template image from a data base.

This process is executed for every template from the data base and the object with the largest similarity is eventually chosen. The steps of the PPMA algorithm are summarized in Fig. 2.

Step 1. Let us assume, that *M* represents the base matrix (component of the database) and *N* – the being recognised one. In both of them 0 means background and 1 – point belonging to the object.
Step 2. For *i* =1, 2, ..., *m*-1, where *m* – number of columns in M, do steps 3 – 6.
Step 3. Increase the number of "*ones*" in *M* – for each element equal to 1 put 1 several times (e.g. $\sigma = 2$) into rows above and below.
Step 4. Do logical *AND* between *M* and *N*. The result put into matrix *I*.
Step 5. Calculate the number of "*ones*" in *I* and put it into vector *MAX* at location *i*.
Step 6. Do circular shift of matrix *M* into right.
Step 7. Find the maximum value in vector *MAX*. It is the value of similarity between *M* and *N*.

Fig. 2. PPMA – Partial Point Matching Algorithm [3]
Rys. 2. PPMA - algorytm częściowego dopasowywania z użyciem punktów [3]

3. Implementation details

We implemented the approach described in Section 2 in SystemC language [6] using CoCentic System Studio from Synopsys[1]. The compared whole images are of the size 64x64, which is too large to execute efficiently in software. (Assuming that every instruction is executed in one cycle, the total execution time of sequential, software implementation would be 139,774 cycles.) In order to increase the speed of the obtained system, we decomposed the template matching block into 16 ones operates on smaller images by 87.5 per cent (the growth of the algorithm is linearly proportional to the area of the image).

The model of the system is presented in Fig. 3. Each of 16 computation units obtains its image line-by-line, one line is transmitted in a single clock cycle using the data bus and the address bus. The most significant bit of the address bus carries the information whether the data of the template or the image to be recognized is transferred. The appropriate data is transferred in 512 clock cycles in case of the comparison with first image in the data base (as the whole image to be recognized is to be transferred), whereas the data required for the next images are transferred in 272 clock cycles.

When all the blocks receive their portion of both images, they start to execute the PPMA algorithm. They compute the similarity degree without cyclic shift operation and transmits it to the *Adder* block (second from the right in Fig. 3). Then, the data is shifted in every image, whereas the lacking points are sent by the *Ctrl* block (third from the left in Fig. 3) in 16 cycles. The operation is repeated 64 times.

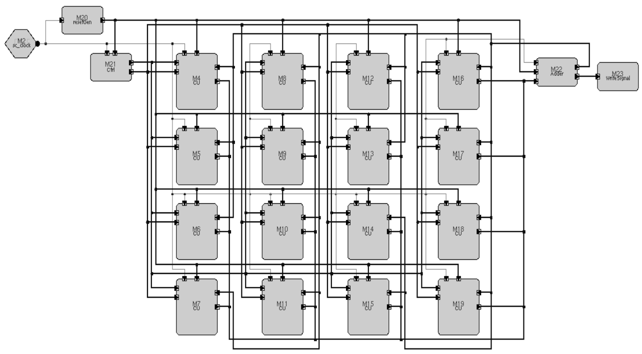


Fig. 3. Model of the PPMA hardware implementation
Rys. 3. Model implementacji sprzętowej algorytmu PPMA

Then, we synthesized one computation unit. Its hardware implementation size was 7,498 LUTs in Spartan 3 FPGA chip for a single computation unit, and the computation time of a single step was equal to 1012.7 ns according to an accurate estimation from a cycle accurate simulator (with the -4 speed grade option), that takes into consideration both logic gate and routing delays. Notice that this area is so large that the implementation of the whole system would not fit into any of Spartan 3 devices (having up to 74,880 LUTs), but it was obtained with the full loop unrolling (FLU) option [1]. After altering the synthesis of the most costly loop into partial loop unrolling, we obtained 2,923 LUTs, i.e., the value that together with the remaining two blocks will fit into XC3S5000 and even XC3S4000. Then, assuming a cycle clock 1 GHz in software, a single hardware implementation of the computation unit (with FLU) computes its data almost 9 times faster that its software counterpart where both of these units works with 16x16 image area. However, when we take into account the parallel hardware vs. sequential software case, we obtain a speed-up equal to two orders of magnitude. This large difference was caused with the facts that (1) the hardware blocks work on much smaller areas and (2) the algorithm is comprised of several loops iterating for every pixel of the image. As these iterations are not dependent on each other, the loops were unrolled and executed in parallel. Unfortunately, the used estimator does not allow us to measure the implementation with partial loop unrolling, but its speed may be maximally 16 times slower than the FLU approach, so it remains almost 10 times faster using pessimistic assumptions.

4. Conclusion

In this paper, we have presented the hardware model of Partial Point Matching Algorithm (PPMA) written in SystemC and presented some data gathered by a cycle accurate simulator of its area and execution time.

The provided experimental results have presented that PPMA is a feasible algorithm for hardware implementations due to the possibility of (1) decomposing the computation into a number of independent blocks (in our case we have used 16 computational

[1] The Software described in this document is furnished under a license from Synopsys (Northern Europe) Limited

units) and (2) the possibility of full loop unrolling of all the loops from the algorithm due to the independence of their iterations. The implementation with all the loops utterly unrolled is more than 100 times faster than a software implementation executed by a processor with 1 GHz clock (after some basic assumptions), but is too large to be implemented in any chip from the Spartan 3 FPGA family. Despite an applying the partial loop unrolling approach worsen the speed-up to one order of magnitude (using pessimistic assumptions), it fits into the second largest Spartan 3 FPGA chip.

Synopsys and the Synopsys product names described herein are trademarks of Synopsys, Inc.

5. References

[1] A. Elbirt, C Paar, An FPGA Implementation and Performance Evaluation of the Serpent Block Cipher, Proceedings of the 8th ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Monterey, USA, 2000, pp. 33-40.

[2] D. Frejlichowski, License plate recognition with significantly distorted characters, Polish Journal of Environmental Studies, vol. 15, no. 4c, 2006, pp. 42-45.

[3] D. Frejlichowski, Trademark retrieval in the presence of occlusion. In: Klopotek M.A., Wierzchon S.T., Trojanowski K. (eds.): Intelligent Information Processing and Web Mining. Advances in Soft Computing, Vol. 35, Berlin Heidelberg, 2006, pp. 253-262.

[4] S. Loncaric, A survey on shape analysis techniques, Pattern Recognition, vol. 31, no. 8, 1998, pp. 983 – 1001.

[5] G. Kuchariew, Digital Image Processing and Analysis (in Polish), Szczecin University of Technology Press, 1998.

[6] OSCI, SystemC Version 2.0 User's Guide, www.systemc.org, 2002.

Artykuł recenzowany

INFORMACJE

Studia Podyplomowe

Wydział Elektryczny Politechniki Śląskiej w Gliwicach, Instytut Metrologii, Elektroniki i Automatyki
ogłasza nabór na Dwusemestralne Zaoczne Studia Podyplomowe

Organizacja i Akredytacja Laboratoriów

Cel studiów

Celem studiów jest pogłębienie wiedzy w zakresie systemu jakości laboratoriów wzorcujących, problematyki zapewnienia jakości wyposażenia pomiarowego, walidacji metod pomiarowych, metodyki tworzenia budżetów niepewności i opracowania wyników badań zgodnie z obowiązującymi przepisami oraz przygotowanie słuchaczy do samodzielnej pracy w zakresie organizowania i prowadzenia laboratorium akredytowanego. Przedstawione zostaną podstawy automatyzacji pomiarów i organizacji systemów pomiarowych. Problemy analizowane będą na przykładach, z uwzględnieniem niezbędnych podstaw teoretycznych oraz aktualnych przepisów.

Profil uczestnika studiów

Studia przeznaczone są dla pracowników o różnych specjalnościach, zajmujących się organizacją laboratoriów oraz wykonywaniem badań i kalibracji w zakładach, firmach lub jednostkach naukowo-badawczych. Studia adresowane są do osób z wyższym wykształceniem zajmujących się realizacją pomiarów i opracowywaniem wyników badań w różnych dziedzinach. Ich ukończenie pozwoli uczestnikom na podwyższenie kwalifikacji niezbędnych do efektywnego opracowywania i dokumentowania procesów pomiarowych. Absolwent studiów otrzymuje Świadectwo Ukończenia Studiów Podyplomowych w zakresie objętym nazwą studiów.

Studia prowadzone są na Wydziale Elektrycznym Politechniki Śląskiej w Gliwicach, w systemie zaocznym w każdą sobotę lub w co drugi weekend (do wyboru) przez dwa semestry. Zajęcia prowadzone są przez nauczycieli akademickich ze stopniem co najmniej doktora oraz przez zaproszonych Gości o uznanym dorobku i autorytecie. Studia obejmują 200 godzin dydaktycznych. Rozpoczęcie Studiów nastąpi po skompletowaniu odpowiedniej liczby kandydatów na dany rodzaj studiów.

Warunki przyjęcia na studia:

1. Na studia mogą być przyjęte osoby posiadające dyplom magistra lub inżyniera, posiadające podstawową wiedzę z zakresu wybranych studiów.
2. Warunkiem uruchomienia studiów jest przyjęcie odpowiedniej liczby Kandydatów na podstawie złożonych dokumentów.
3. Dokumenty składane przez Kandydatów:
   - Kwestionariusz Osobowy – Karta Zgłoszenia (do pobrania ze strony internetowej). Przyjmowane na bieżąco: e-mailem, pocztą lub osobiście.
   - Kopia/odpis dyplomu ukończenia studiów wyższych.
4. Kandydaci odbywają rozmowę kwalifikacyjną. Termin ustalony i podany zostanie po skompletowaniu odpowiedniej liczby Kandydatów.
5. Po spełnieniu warunków Kandydaci wnoszą opłatę zgodnie z zawartą umową w wysokości 3 800 złotych za cały okres studiów.

Organizator studiów:

Instytut Metrologii, Elektroniki i Automatyki Politechniki Śląskiej, 44-100 Gliwice, ul. Akademicka 10, tel. 032 237 12 41, fax: 032 237 20 34, e-mail: re2@polsl.pl lub agnieszka.skorkowska@polsl.pl, http://www.wega.elektr.polsl.gliwice.pl

Kierownik studiów:

Prof. dr hab. inż. Tadeusz SKUBIS