A Parallel String Search Algorithm

Yoshiyasu Takefuji, Toshimitsu Tanaka, and Kuo Chun Lee

Abstract— **A new parallel processing algorithm for solving string search problems is presented in this paper. The proposed algorithm uses $O(m \times n)$ processors where n is the length of a text and m is the length of a pattern. It requires two and only two iteration steps to find the pattern in the text, while the best existing parallel algorithm needs the computation time $O(\text{loglog } n)$.**

I. INTRODUCTION

The string search problem is to find all occurrences of a given m-character pattern in a given n-character text and it is one of fundamental operations in information system and computer science. As is well known, one of the best current string search algorithm was proposed by Boyer–Moore [1]. Their algorithm performs in $O(n/m)$ on the average case. However, the performance of their algorithm is poor for short patterns when m is small, and the worst case of their algorithm is $O(n+rm)$ where r is the number of matches found [2], [3]. Another well known Knuth, Morris, and Pratt's algorithm [4] is also $O(n+m)$ in the worst case.

A few parallel algorithms for solving string search problems have been proposed in the last decade. In 1984, Galil proposed optimal parallel algorithms where one algorithm requires $O(k)$ time with $p = n^{1+1/k}$ and the other does $O(\log n/\text{loglog } n)$ time with $p = n$ [5]. In 1985, Vishkin proposed the $O(n/p)$ time parallel algorithm with $pn/\log n$ [6]. Berkman, Breslauer, Galil, Schieber, andViskin presented the $O(\text{loglog } n)$ time parallel algorithm in 1989 [7]. String-search VLSI circuits have been investigated by Hirata [8], Foster [9], and Wade [10]. In this paper, a new parallel algorithm to find all occurrences of a pattern among the given text is presented. The algorithm uses simple $m \times n$ processing elements called binary neurons and requires two and only two iteration steps to find a solution. A super parallel sorting algorithm was reported in [11], [12] where they use the same McCulloch–Pitts neuron. The output of the ith McCulloch–Pitts binary neuron is given by

$$V_i = f(U_i) = \begin{cases} 1, & \text{if } U_i > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where V_i is the output of the ith neuron and U_i is the input to the ith neuron.

In 1985 the first artificial neural network for optimization problems was introduced by Hopfield and Tank [13]. Although Wilson and Pawley [14] and Paielli [15] strongly criticized the neural network for optimization problems, a variety of optimization problems including sorting [11], [12], graph planarization [16], tiling [17], RNA secondary structure prediction [18], [19], finding a maximum independent set [19], crossbar switch scheduling [20], time slot assignment problems in TDM hierarchical switching systems [21], channel routing [22], Hip games [23], four-coloring and k-colorability

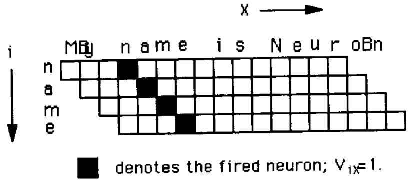


Fig. 1. Neural network representation for string search problems: $M = 4$ and $n = 18$.

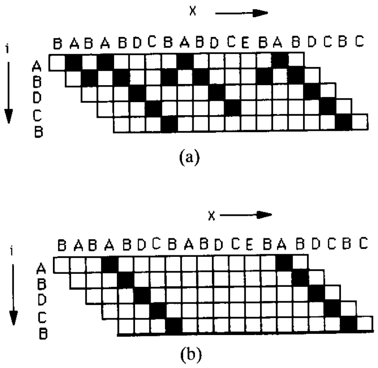
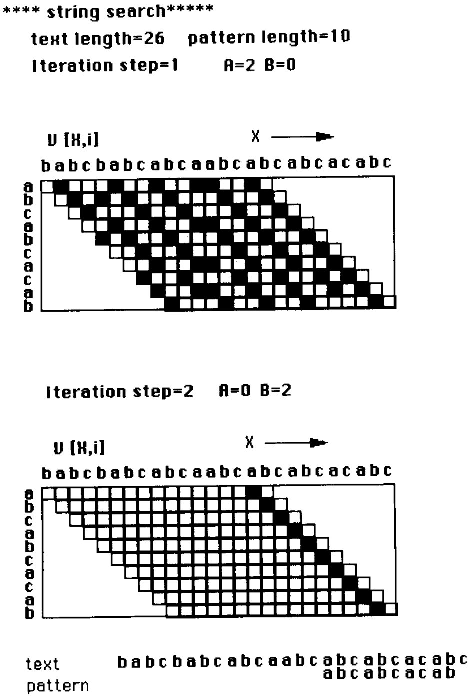


Fig. 2. States of neurons. Black box denotes the fired neuron: $m = 5$, $n = 20$. ■ denotes the final neutron. (a) A state after the first step. (b) A state after the second step.



```
**** string search*****
    text length=26   pattern length=10
    Iteration step=1        A=2 B=0

    Iteration step=2   A=0 B=2

text      babcbabcabcaabcabcabcacabc
pattern               abcabcacab
```

Fig. 3. The simulation result of the text size=26, pattern size= 10.

Manuscript received September 19, 1990; revised May 22, 1991.

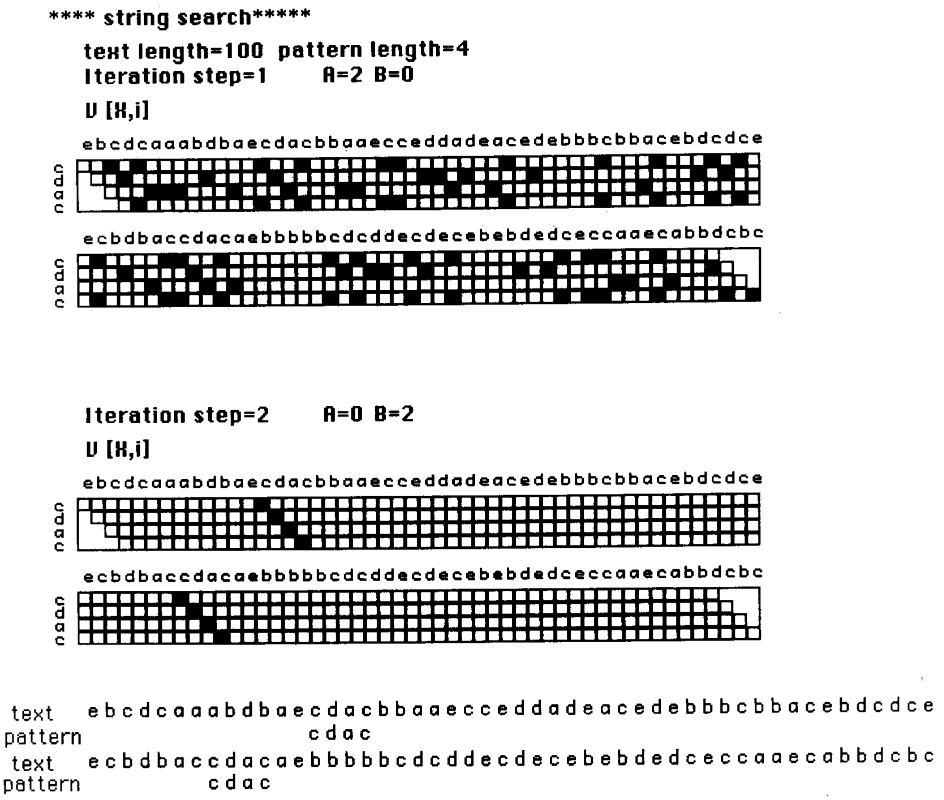
Y. Takefuji is with the Department of Electrical Engineering, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland, OH 44106, and also with the Faculty of Environmental Information, Keio University, Fujisawa 252, Japan.

K. C. Lee is with the Department of Electrical Engineering, Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland, OH 44106.

T. Tanaka is with Sumitomo Metal Industries Ltd., Kashima Steel Works, Ibaragi, 314 Japan.

IEEE Log Number 9104102.

0018-9472/92$03.00 © 1992 IEEE



```
**** string search*****
    text length=100  pattern length=4
    Iteration step=1      A=2 B=0
    U [X,i]
    ebcdcaaabdbaecdacbbaaecceddadeacedebbbcbbacebdcdce

    ecbdbaccdacaebbbbbcdcddecdecebebdedceccaaecabbdcbc


    Iteration step=2      A=0 B=2
    U [X,i]
    ebcdcaaabdbaecdacbbaaecceddadeacedebbbcbbacebdcdce

    ecbdbaccdacaebbbbbcdcddecdecebebdedceccaaecabbdcbc


text    e b c d c a a a b d b a e c d a c b b a a e c c e d d a d e a c e d e b b b c b b a c e b d c d c e
pattern                           c d a c
text    e c b d b a c c d a c a e b b b b b c d c d d e c d e c e b e b d e d c e c c a a e c a b b d c b c
pattern               c d a c
```

Fig. 4. The simulation result of the text size=100, pattern size=4.

[24], spare allocation problems [25], and others [26] have been successfully solved. In this paper, the neural network representation is given in Section II, the simulation result in Section III, the hardware architecture in Section IV, and the conclusion in Section V.

II. Neural Network Representation

An $m(n-m+1)$ neural network array is provided for solving a string search problem where row and column correspond to the location of the pattern register and that of the text register, respectively. U_{ix} and V_{ix} represent an input and an output of the ixth neuron, respectively, where the subscript i indicates the ith register which contains the ASCII value of the ith character in the pattern, N_{pi}. In the same way, the second subscript x indicates the xth register which contains the ASCII value of the xth character in the text, N_{tx}. If more than one pattern exist in the given text, the corresponding ixth neuron will be fired, that is, $V_{ix} = 1$. For example, Fig. 1 shows the neural network representation for searching the four-character pattern "name" in the 18-character text "My name is Neuron." Fired neurons locate the string of the searching pattern.

The motion equation of the ixth neuron is given by

$$\frac{dUix}{dt} = -A \cdot g(N_{pi} - N_{tx}) - B \cdot g\left(\sum_{j=1}^{m} V_{jX-i+j} - m\right) \qquad (2)$$

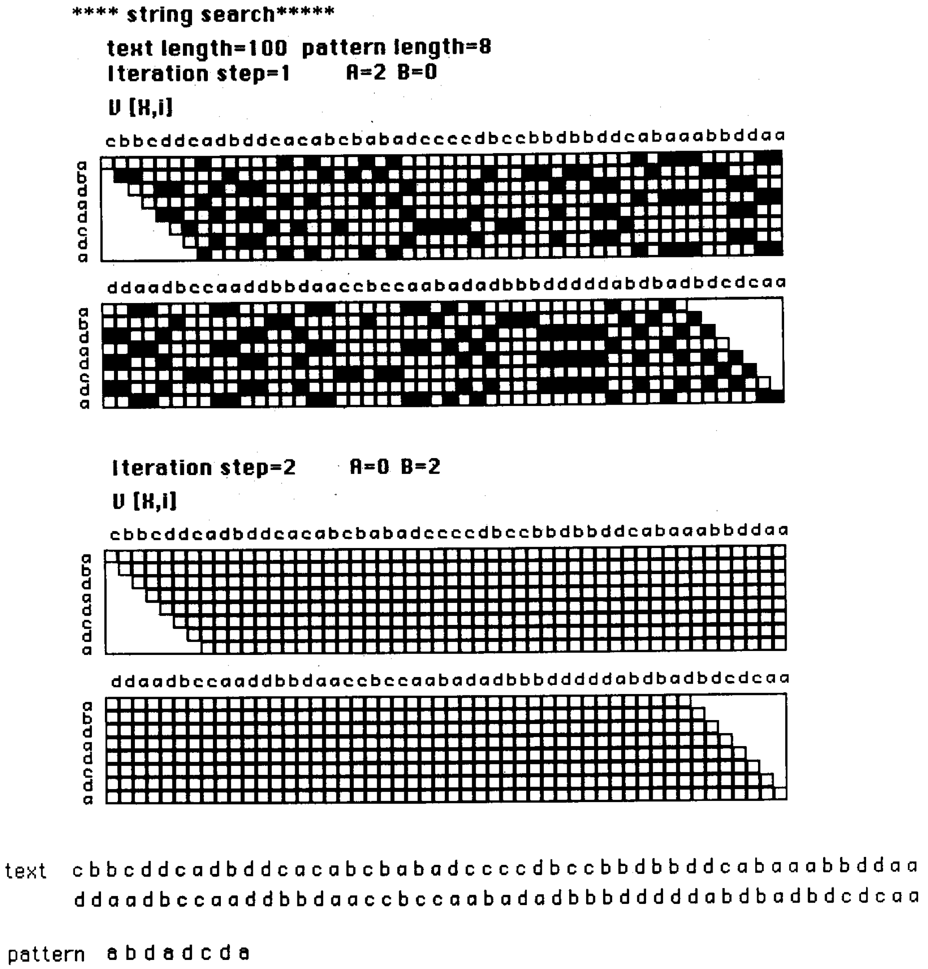
where the function $g(x)$ is -1 if $x = 0, 1$ otherwise. A and B are coefficients and m is the length of a pattern. In (2), the first term performs the excitatory forces as long as the ith character in the pattern is the same as the xth character in the text. If $N_{pi} = N_{tx}$ then V_{ix} will be 1, else the first term will act as the inhibitory force. The second term will be the excitatory force when diagonally consecutive m neurons including the ixth neuron are all fired. If not, the second term will discourage the ixth neuron to fire.

The first-order Euler method was used for the numerical simulation of (2) and each simulation run was terminated when the patterns were found in the given text. Initial values of neurons U_{ix} for $i = 1, \cdots, m$ and $x = 1, \cdots, n$ were assigned to the same negative value. The following procedure describes the proposed parallel algorithm.

1) Set $t = 0$.
2) The initial values of $U_{ix}(0)$ for $i = 1, \cdots, mx = 1, \cdots, n$ are assigned to the same negative values. For example $U_{ix}(0) = -1$for $i = 1, \cdots, mandx = 1, \cdots, n$.
3) Evaluate the values of $V_{ix}(t)$ based on the McCulloch–Pitts binary function. If $U_{ix}(t) > 0$ then $V_{ix}(t) = 1$, else $V_{ix}(t) = 0$
4) Use the motion equation in (2) to compute $U_{ix}(t)$. In the first iteration step, the coefficient A is assigned to 2 and B is assigned to 0. In the second iteration step, A is changed to 0 and B is changed to 2:

$$\Delta U_{ix}(t) = -A \cdot g(N_{pi} - N_{tx}) - B \cdot g\left(\sum_{j=1}^{m} V_{jx-i+j}(t) - m\right). \qquad (3)$$

5) Compute $U_{ix}(t+1)$ based on the first-order Euler method: $U_{ix}(t+1) = U_{ix}(t) + U_{ix}(t)$ for $i = 1, \cdots, m$ and $x = 1, \cdots, n$.
6) Increment t by 1.
7) If $t > 2$ then terminate this procedure, else go to step 2.



```
**** string search*****
  text length=100  pattern length=8
  Iteration step=1      A=2 B=0
  U [X,i]
  cbbcddcadbddcacabcbabadccccdbccbbdbbddcabaaabbddaa

  ddaadbccaaddbbdaaccbccaabadadbbbdddddabdbadbdcdcaa

  Iteration step=2      A=0 B=2
  U [X,i]
  cbbcddcadbddcacabcbabadccccdbccbbdbbddcabaaabbddaa

  ddaadbccaaddbbdaaccbccaabadadbbbdddddabdbadbdcdcaa

text    c b b c d d c a d b d d c a c a b c b a b a d c c c c d b c c b b d b b d d c a b a a a b b d d a a
        d d a a d b c c a a d d b b d a a c c b c c a a b a d a d b b b d d d d d a b d b a d b d c d c a a

pattern  a b d a d c d a
```

Fig. 5. The simulation result of the text size=100, pattern size= 8.

III. SIMULATION RESULTS

Fig. 2(a) shows the state of the neurons after the first iteration step. At this stage, every ixth neuron whose N_{pi} equals to N_{tX} is fired because the first A term of (3) is positive and the second term is zero. After the second step, only diagonally consecutive m neurons among fired neurons are forced to remain fired as shown in Fig. 2(b). At the second step, the second term of (3) encourages the exact matching patterns to remain while the first term is zero.

A large number of simulation runs were performed with various length of texts and patterns on Macintosh SE/30. Figs. 3–5 show the simulation results of the text length 26, 100, and 100, respectively. The text and the pattern data given in [4] were used in Fig. 3. The characters of the text and the patterns in Figs. 4 and 5 were randomly generated. In Fig. 4, the pattern "cdac" is located at two places in the 100-length text. Fig. 5 shows the example that the pattern " abdadcda" does not exist among the given text.

IV. ARCHITECTURE

Fig. 6 depicts the architecture of the proposed parallel string-search system based on (2) where it is composed of $m(n-m+1)$ processing elements. S_{AiX} and S_{iX} represent circuits that consist of switches, comparators, and a summing operator. Fig. 7 shows the detailed circuit diagram of a neuron in Fig. 6. S_a is a switch which turns on at the first step and off at the second step. S_b turns off at the first step and on at the second step. S_{AiX} and S_{BiX} are single-pole–double-throw switches which select the input value for the iXth neuron according to the result of the comparator operation. The comparator C_{Aix} compares two input values, N_{pi} andN_{tX}. If they are equal then S_{AiX} changes the connection to the A-line so that the output of S_{AiX} will be the value of the coefficient A. If N_{pi} and N_{tX} are not equal then S_{AiX} does not change and the output will be $-A$. Σ_{ix}denotes the summing operator that implements the summation term in (2). The output of this operator is one of the input to the comparator C_{Bix}. Fig. 8 shows the circuit of the summing operator. Fig. 9 describes the implementation of a neuron using analog operational amplifiers where the circuit is not minimized yet for the actual implementation. The first operational amplifier in Fig. 9 performs integration that is given by the following equation:

$$-\frac{1}{CR}\int\left(\frac{dU_{iX}}{dt}\right)dt = -\frac{U_{iX}}{CR}. \tag{4}$$

The second operation amplifier generates U_{ix}:

$$-\left(-\frac{U_{iX}}{CR}\right)R'U_{iX}. \tag{5}$$

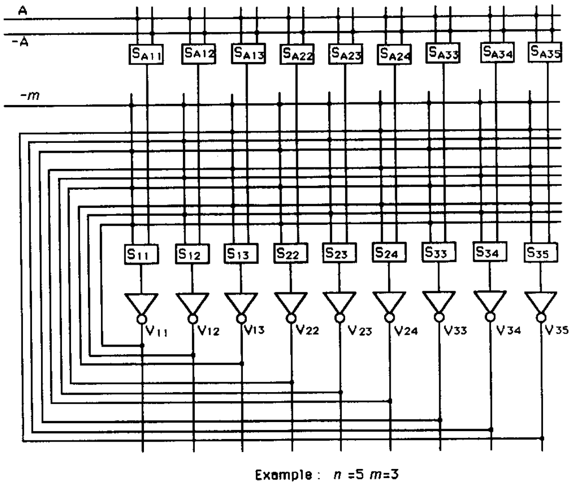


Fig. 6. A circuit diagram of the string search neural network.

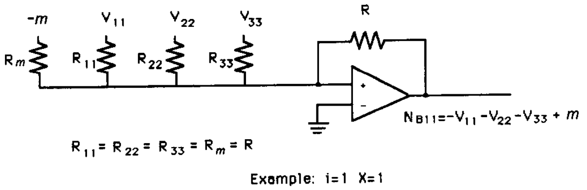


Fig. 8. A summing operation circuit.

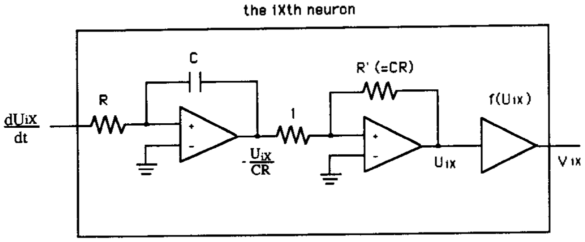


Fig. 9. An analog circuit of the ixth neuron.

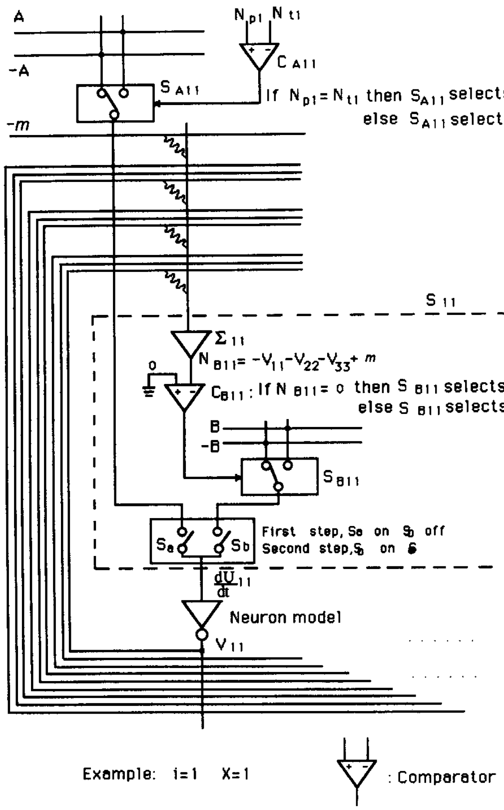


Fig. 7. A detail diagram of one neuron.

The last component in Fig. 9 is a binary function determined by (1). It is also possible to implement this algorithm using digital circuits.

V. Conclusion

Based on our simulation result, the proposed algorithm for solving the string search problems was verified. It shows the algorithm's consistency. It finds a solution with two and only two iteration steps, regardless of the problem size. Our algorithm provides simultaneous search for more than one patterns in the same text. The algorithm requires $m(n - m + 1)$ processing elements and $2m(n - m + 1)$ comparators.

References

[1] R. S. Boyer and J. S. Moore, "A fast string searching algorithm," *Commun. Ass. Comput. Mach.*, vol. 20, pp. 762–772, 1977.
[2] G. D. Smit, "A comparison of three string matching algorithms," *Software Practice and Experience*, vol. 12, pp. 57–66, 1982.
[3] R. A. Baeza-Yates, "Improved string searching," *Software Practice and Experience*, vol. 19, pp. 257–271, 1989.
[4] D. E. Knuth, J. H. Morris, Jr., and V. B. Pratt, "Fast pattern matching in strings," *SIAM J. Comput.*, vol. 6, pp. 323–350, 1977.
[5] Z. Galil, "Optimal parallel algorithms for string matching," in *Proc. 16th ACM Symp. Theory of Computing*, 1984, pp. 240–248.
[6] U. Vishkin, "Optimal parallel pattern matching in strings," in *Proc. 12th ICALP, Lecture Notes in Computer Science*, vol. 194, 1985, pp. 497–507.
[7] O. Berkman, D. Breslauer, G. Galil, B. Schieber, and U. Vishkin, "Highly parallelizable problems," in *Proc. 21st ACM Symp. Theory of Computing*, 1989.
[8] M. Hirata, H. Yamada, H. Nagai, and K. Takahashi, "A versatile data string-search VLSI," *IEEE J. Solid-State Circuits*, vol. 23, pp. 329–335, 1988.
[9] M. J. Foster and H. T. Kung, "The design of special-purpose VLSI chips," *IEEE Computer*, vol. 13, pp. 26–40, 1980.
[10] J. P. Wade, P. J. Osler, R. E. Zippel, and C. G. Sodini, " The MIT database accelerator: 2-K TRIT circuit design," in *Proc. VLSI Symp. Dig. Tech.*, 1987, pp. 39–40.
[11] Y. Takefuji and K. C. Lee, " A super parallel sorting algorithm based on neural networks," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1425–1429, Nov. 1990.
[12] ——, "A two-step parallel sorting algorithm based on neural networks," *J. Neural Network Comput.*, vol. 2, no. 1, pp. 30–32, Summer 1990.
[13] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.
[14] G. V. Wilson and G. S. Pawley, "On stability of the travelling salesman problem algorithm of hopfield and tank," *Biol. Cybern.*, vol. 58, pp. 63–70, 1988.
[15] R. A. Paielli, "Simulation tests of the optimization method of Hopfield and Tank using neural networks," NASA Tech. Memo. 101047, 1988.
[16] Y. Takefuji and K. C. Lee, "A near-optimum arallel planarization algorithm," *Science*, vol. 245, pp. 1221–1223, Sept. 1989.
[17] ——, "A parallel algorithm for tiling problems" *IEEE Trans. Neural Networks*, vol. 1, pp. 143–145, 1990.
[18] Y. Takefuji, C. W. Lin, and K. C. Lee, "A parallel algorithm for estimating the secondary structure in ribonucleic acids," *Biol. Cybern.*,

vol. 63, no. 5, pp. 337–340, 1990.
[19] Y. Takefuji, L. L. Chen, K. C. Lee, and J. Huffman, "Parallel algorithms for finding a near-maximum independent set of a circle graph," *IEEE Trans. Neural Networks*, vol. 1, pp. 263–267, 1990.
[20] Y. Takefuji and K. C. Lee, "An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics," *Biol. Cybern.*, vol. 64, pp. 353–356, 1991.
[21] N. Funabiki and Y. Takefuji, "A parallel algorithm for time slot assignment problems in TDM hierarchical switching systems," *IEEE Trans. Commun.*, to be published.
[22] ——, "A parallel algorithm for channel routing problems," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 464–474, Apr. 1992.
[23] ——, "A parallel algorithm for solving the Hip games," *Neurocomputing*, vol. 3, pp. 97–106, 1991.
[24] Y. Takefuji and K. C. Lee, "Artificial neural networks for four-coloring problems and k-colorability problems," *IEEE Trans. Circuits and Systems*, vol. 38, pp. 326–333, 1991.
[25] N. Funabiki and Y. Takefuji, "A parallel algorithm for spare allocation problems," *IEEE Trans. Reliability*, vol. 40, pp. 338–346, 1992.
[26] Y. Takefuji, *Neural Network Parallel Computing.* Amsterdam, The Netherlands: Kluwer-Academic, Jan. 1992.

Knowledge-Guided Visual Perception of 3-D Human Gait from a Single Image Sequence

Zen Chen and Hsi-Jian Lee

Abstract—**A computer vision method is presented to determine the 3-D spatial locations of joints or feature points of a human body from a film recording the human motion during walking. The proposed method first applies the geometric projection theory to obtain a set of feasible postures from a single image, then it makes use of the given dimensions of the human stick figure, physiological and motion-specific knowledge to constrain the feasible postures in both the single-frame analysis and the multi-frame analysis. Finally a unique gait interpretation is selected by an optimization algorithm. Computer simulations are used to illustrate the ideas presented.**

I. INTRODUCTION

In the past a large amount of work has been devoted to problems of human locomotion, notably walking [1]–[3]. In the human gait analysis the entire body motion during walking is represented as a set of spatial trajectories of joints (or anatomic points) [4]–[7]. The mechanics of joint forces and moments is characterized by angular accelerations, velocities and displacements [2], [8]–[9]. Typical application fields of the gait analysis include the physical therapy of joint diseases, biomechanical simulations, kinesiological analysis and mobile robot design, etc. [2], [10]–[11].

There are two major vision methods: stereo vision and monocular vision. In the stereo vision at least two views of the subject are simultaneously taken, then a triangulation method is applied to these views to compute the 3-D coordinates for those joints appearing in two views simultaneously [12]–[13]. On the other hand, the monocular vision can determine the 3-D motion and structure (unique up to a scaling factor) of the subject based on a number of consecutive frames [14]–[17]. Both approaches have their own advantages and shortcomings [18].

In the human gait analysis, the stereo vision can determine the joint positions without using any *a priori* knowledge. Since the triangulation method completely relies on the two vectors defined by the viewpoints and projected points, any digitization error of projected points will lead to an inaccurate joint position. This is true especially when the two vectors are nearly parallel to each other. As a consequence, the obtained joint positions may not represent the legal (i.e., original) human body model. Furthermore, it is difficult to use any knowledge about the human model to refine the result in the stereo vision method. Therefore, for the well-constrained human body model, the stereo vision may not be suitable.

As to the monocular vision, the method requires a sufficient number of joints on the subject to appear in consecutive frames. It is generally impossible to have so many points for human body segments such as arms and legs. Besides, only the structure, unique up to a scaling factor, can be obtained instead of the exact body position. In the field of the robot vision there are methods that can directly determine the 3-D locations of the subject, if the dimensions of the subject is known beforehand [19]–[20]. However, in these methods some viewing conditions or object structure conditions are assumed; it is not very realistic in the human gait analysis. So far there have been only partial solutions to the visual interpretation problem of the general human motion data [17], [21]–[22].

Up to now only geometrical and topological models of a human body are employed in the gait interpretation which generally lead to nonunique joint position recovery from the film. Rashid [21] indicated that the object topology and world knowledge are required to help the interpretation. Herman [23] tried to obtain a meaningful description of a human body motion while playing baseball by using domain-dependent knowledge about the body model. O'Rourke and Badler [22] used constraints of the human body model such as distance constraints, joint angle limits, collision avoidance to refine the 3-D joint positions. In a previous study, we also used physiological and motion specific constraints to derive a small set of feasible body postures for a single frame [7]. Therefore, the application of various sources of knowledge will reduce the joint position ambiguity and can lead to a small set of candidate solutions.

It is not very meaningful to describe a human motion with only a single frame. Instead, the human motion is better described by a collection of consecutive frames as a whole. Hence, certain candidate solutions obtained in the single-frame analysis may be ruled out by checking the interframe compatibility or consistency pertinent to the motion analysis.

In this study a computer vision method for interpreting the human motion during walking is presented. In Section II basic analyses for gait interpretations are described which lead to a set of possible interpretations. Then a computational model based on a graph search theory is formulated for finding a unique interpretation solution in Section III. Algorithm A^* with a proper evaluation function is proposed to find the solutions. Two sets of experimental data are used in the simulation. The algorithm and simulation results are given in Section IV. The results indicate that the algorithm has some minor defects. In Section V two modifications are made to Algorithm A^*. After these changes, together with the aid of additional motion-specific knowledge, a final unique gait interpretation is reached. The simulation results show the goodness of the method. Section VI gives the conclusion.

Manuscript received March 31, 1989; revised September 8, 1989, and May 22, 1991.

The authors are with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, 30050 R.O.C.

IEEE Log Number 9104117.