JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 16, 117-126 (2000)
DOI:10.1688/JISE.2000.16.1.7

Short Paper

A Parallel Approach for Embedding Large Pyramids Into Smaller Hypercubes With Load Balancing*

YU-WEI CHEN AND KUO-LIANG CHUNG+
Department of Information Science
Tamsui Oxford University College
Tamsui, Taipei County, Taiwan 251, R.O.C.
E-mail: ywchen@jupiter.touc.edu.tw
+*Department of Information Management and Institute of Information Engineering*
National Taiwan University of Science and Technology
Taipei, Taiwan 106, R.O.C.
E-mail: klchung@cs.ntust.edu.tw

With dilation 2, congestion 2, expansion 3/2, and load 1, this short paper first presents a parallel method for embedding a pyramid with height n, P_n for $n \geq 2$, into a $(2n - 1)$ – dimensional hypercube, H_{2n-1}, in $O(n)$ time. With dilation 2, congestion $2^{n-t} + 3$ (or $2^{n-t+1} + 2$), and load $\lceil 2^{2n-k} / 3 \rceil$ when $0 \leq k = 2t$ (or $k = 2t - 1) \leq 2n - 2$, our proposed parallel method is further extended to map P_n into H_k with load balancing in $O(k)$ time.

Keywords: congestion, dilation, parallel algorithm, hypercube, load balancing, nCUBE 2S, pyramid

1. INTRODUCTION

The pyramid [1] is a well-known parallel network in the field of image processing and pattern recognition [2, 3]. The hypercube is one of the most versatile and popular networks because it can efficiently simulate many different networks [4, 5]. Previously, some one-to-one embedding methods [3, 5, 6] have been presented to map pyramids into hypercubes where one hypercube node emulates at most one pyramid node. With dilation 2, expansion 3/2, and load 1, Stout [3] presented an embedding method to map pyramids into hypercubes. Because the paths, which are used to emulate the pyramid edges, in the hypercubes are not specified, Lai and White [6] claimed that the congestion in Stout's methods [3] is uncertain. Although Stout [3] did not consider congestion, Monien and Sudborough [5] claimed that the congestion in Stout's method is 2. With dilation 2, congestion 2, expansion 3/2, and load 1, Monien and Sudborough [5] presented a recursive embedding method to map pyramids

Received March 19, 1998; revised March 4, 1999; accepted April 1, 1999.
Communicated by Jang-Ping Sheu.
* This research was supported by the National Science Council of R.O.C. under contract NSC87-2213-E011-001.

into hypercubes. With dilation 3 (2), congestion 2 (3), expansion 3/2, and load 1, Lai and White [6] presented one (the other) recursive embedding method to map pyramids into hypercubes.

With dilation 2, congestion 2, expansion 3/2, and load 1, this short paper first presents a parallel method for embedding a pyramid with height n, P_n for $n \geq 2$, into a $(2n-1)$-dimensional hypercube, H_{2n-1}, in $O(n)$ time. Not only is the proposed parallel method quite different from those in [6], but it also has smaller dilation and congestion when compared to the first method and the second method presented in [6], respectively. The difference between the proposed embedding method and the two methods in [6] will be discussed in section 3. In practice, we may find that the size of the hypercube is smaller than that of the pyramid. Thus, it is necessary to develop an efficient method to map large pyramids into smaller hypercubes. With dilation 2, congestion $2^{n-t}+3$ (or $2^{n-t+1}+2$), and load $\lceil 2^{2n-k}/3 \rceil$ when $0 \leq k = 2t$ (or $k = 2t-1$) $\leq 2n-2$, our proposed parallel method is further extended to map P_n into H_k in $O(k)$ time. Because all the nodes in P_n are evenly embedded into 2^k nodes in H_k, each node in H_k has the same load. This is why the proposed embedding method has the load-balancing capability. To the best of our knowledge, this is the first time such a parallel embedding method with load balancing has been proposed in the literature.

The remainder of this short paper is organized as follows. The next section describes some basic definitions and terminologies. Section 3 presents the first parallel method for embedding P_n into H_{2n-1}. Section 4 presents a parallel method for mapping P_n into H_k, $0 \leq k \leq 2n-2$, with load balancing. Finally, some conclusions are dawn in section 5.

2. DEFINITIONS AND TERMINOLOGIES

We introduce here the pyramid that is being embedded and the hypercube that we are embedding into. We define P_h as the h-level pyramid with vertex set $V(P_h) = \bigcup_{l=0}^{h-1}\{(l, y, x)\colon 0 \leq y, x \leq 2^l - 1)\}$ and edge set $E(P_h) = \bigcup_{l=1}^{h-1}\{((l, y_1, x_1), (l, y_2, x_2))\colon |y_1 - y_2| + |x_1 - x_2| = 1$ and $(l, y_1, x_1), (l, y_2, x_2) \in V(P_h) =\} \cup \bigcup_{l=1}^{h-1}\{((l-1, \left\lfloor \frac{y}{2} \right\rfloor, \left\lfloor \frac{x}{2} \right\rfloor), (l, y, x))\colon 0 \leq y, x \leq 2^l - 1\}$. Fig. 1 shows a P_3. For the purpose of exposition, we call the top node in the pyramid or any one subpyramid the apex node. For example, node (0, 0, 0) in Fig. 1 is the apex node of pyramid P_3. An H_k has 2^k nodes and $k2^{k-1}$ edges, where two nodes are linked with an edge if and only if their binary strings differ by exactly one bit. Two terms, binary string and binary number, are interchangeably used in this short paper. Fig. 2 shows an H_3.

A function f for embedding a graph $G(V, E)$ into a graph $G'(V', E')$ is a mapping of $V(G)$ into $V'(G')$, combined with a mapping of $e = (u, v) \in E(G)$ into a simple path of $G'(V', E')$ so that $f(e) = (f(u), f(v))$ is a simple path of $G'(V', E')$ with end points $f(u)$ and $f(v)$. Commonly, dilation, expansion, congestion, and load are used to evaluate the efficiency of embedding methods. The maximum distance we must stretch any edge to achieve embedding is called the dilation. The expansion denotes the ratio of the number of nodes in the target network to that in the source network, i.e., $\frac{|V'(G')|}{|V(G)|}$. The congestion is the maximum number of edges of the source network that are embedded and share any single edge of the target network. The load is the maximum number of nodes of the source network that are embedded in any single node of the target network.

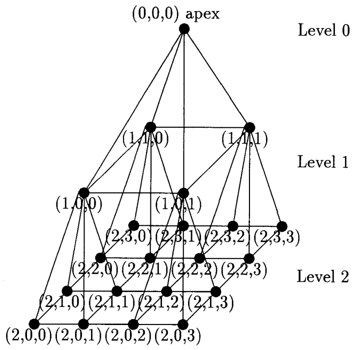

Fig. 1. A pyramid with three levels, P_3.

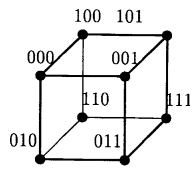

Fig. 2. A three-dimensional hypercube, H_3.

3. EMBEDDING P_n INTO H_{2n-1}

In this section, we will present an embedding function to map P_n into H_{2n-1} with dilation 2, expansion 3/2, congestion 2, and load 1. The initial concept behind our embedding method is that P_2 is first mapped into H_3, then one augmented apex node and four P_2's are mapped into H_5; continuing this way, after mapping P_k into H_{2k-1}, one augmented apex node and four P_k's are further mapped into H_{2k+1}.

Using the binary-reflected Gray codes and the above recursive construction approach, the embedding function f_1 used to map P_n into H_{2n-1} is defined as

$$f_1(l,y,x)=\begin{cases}(G^l(y)G_0^l(x)1^{n-l-2},G^l(x)0^{n-1}), & 0\le l<n-1\\(G^{n-1}(y),G^{n-1}(x)G_0^{n-1}(x)), & l=n-1,\end{cases}$$

where (l, y, x) is the address of one node in P_n and the binary-reflected Gray code $G^n(b) = g_{n-1}g_{n-2}\cdots g_0$ of an n-bit binary string $b = b_{n-1}b_{n-2}\cdots b_0$ is defined as $g_i = b_i \oplus b_{i+1}$, $0 \le i \le n-2$, and $g_{n-1} = b_{n-1}$. Here, the symbol $\oplus$ is defined as a bitwise exclusive-or (XOR) operator. The lengths of $G^l(x)$ and $G_0^l(x)$ are l and 1, respectively, where $G_0^l(x)$ denotes the least significant bit of $G^l(x)$. Note that $G^0(x)$ and 1^0 are empty strings, and that $G_0^0(x)$. The symbols 1^{n-1-2} and 0^{n-1} denote the strings $\underbrace{11\cdots11}_{n-l-2}$ and $\underbrace{00\cdots00}_{n-l}$, respectively. $(G^l(y)G_0^l(x)1^{n-l-2},G^l(x)0^{n-1})$ of f_1 denotes a $(2n-1)$-bit string combined by 5 binary strings with length l, 1, $n-l-2$, l, and $n-1$. Using f_1, embedding P_3 into H_5 is shown in Fig. 3, where the lines in H_5 denote paths emulating the edges in P_3. The apex node (0, 0, 0) in P_3, for example, is embedded into the node 01000 (=(01, 000)) in H_5.

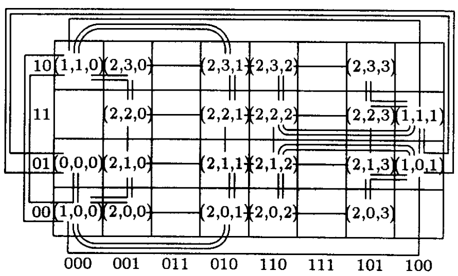

Fig. 3. Embedding P_3 into H_5.

In the following three lemmas, we will show that embedding P_n into H_{2n-1} can be done with load 1, dilation 2, and congestion 2.

Lemma 1. Using the embedding function f_1, embedding P_n into H_{2n-1} has load 1.

Proof: Suppose any two distinct nodes (l_1, y_1, x_1) and (l_2, y_2, x_2) in P_n are embedded into the same node in H_{2n-1}. There are three cases to be considered. For the first case $l_1 = l_2$, by f_1, we have either $(G^{l_1}(y_1)G_0^{l_1}(x_1)l^{n-l_1-2}, G^{l_1}(x_1)0^{n-l_1}) = (G^{l_2}(y_2)G_0^{l_2}(x_2)l^{n-l_2-2}, G^{l_2}(x_2)0^{n-l_2})$, or $(G^{n-1}(y_1), G^{n-1}(x_1)\overline{G_0^{n-1}(x_1)}) = (G^{n-1}(y_2), G^{n-1}(x_2)\overline{G_0^{n-1}(x_2)})$. Because (l_1, y_1, x_1) and (l_2, y_2, x_2) are distinct, we know that $G^{l_1}(y_1) \neq G^{l_2}(y_2)$ or $G^{l_1}(x_1) \neq G^{l_2}(x_2)$. Thus, the two nodes (l_1, y_1, x_1) and (l_2, y_2, x_2) are not embedded into the same nodes, and this is a contradiction. For the second case $l_1 < l_2$ and $l_2 \neq n-1$, by f_1, we have the following two equations: $G^{l_1}(y_1)G_0^{l_1}(x_1)1^{n-l_1-2} = G^{l_2}(y_2)G_0^{l_2}(x_2)1^{n-1_2-2}$ and $G^{l_1}(x_1)0^{n-l_1} = G^{l_2}(x_2)0^{n-l_2}$. It follows that $G^{l_1}(y_1)G_0^{l_1}(x_1)1^{l_2-l_1} = G^{l_2}(y_2)G_0^{l_2}(x_2)$ and $G^{l_1}(x_1)0^{l_2-l_1} = G^{l_2}(x_2)$. Since $l_1 - l_2 > 0$, we have $G_0^{l_2}(x_2) = 1$ and $G_0^{l_2}(x_2) = 0$, and this is a contradiction. For the third case $l_1 < l_2$ and $l_2 = n-1$, two nodes (l_1, y_1, x_1) and (l_2, y_2, x_2) are embedded into two nodes in subcubes $\{0, 1\}^{2n-3}00$ and $\{0, 1\}^{2n-3}G_0^{n-1}(x)\overline{G_0^{n-1}(x)}$, respectively. This is a contradiction. As a result, embedding P_n into H_{2n-1} by using f_1 has load 1. □

Lemma 2. Using the embedding function f_1, embedding P_n into H_{2n-1} has dilation 2.

Proof: Let the function $dist(p, q)$ denote the Hamming distance between two binary strings p and q. We will first consider the edge linking of any node $(l+1, y, x)$ to its parent node $\left(l, \left\lfloor \frac{y}{2} \right\rfloor, \left\lfloor \frac{x}{2} \right\rfloor\right)$, $0 \leq l < n-1$. By the definition of Gray code, $G^l\left(\left\lfloor \frac{y}{2} \right\rfloor\right)$ is the same as the leftmost l bits of $G^{l+1}(y)$, e.g., $G^3(5) = 111$ and $G^4(10) = 1111$. There are two cases to be considered: $0 \leq l < n-2$ and $l = n-2$. The dilations for the two cases, $0 \leq l < n-2$ and $l = n-2$, are given by $dist(f_1(l+1, y, x), f_1(l, \left\lfloor \frac{y}{2} \right\rfloor, \left\lfloor \frac{x}{2} \right\rfloor)) =$ and $dist(G_0^{l+1}(y), G_0^l(\left\lfloor \frac{x}{2} \right\rfloor)) + dist(G_0^{l+1}(x), 1) + dist(G_0^{l+1}(x), 0) \leq 2$ and $dist(f_1(n-1, y, x), f_1(n-2, \left\lfloor \frac{y}{2} \right\rfloor, \left\lfloor \frac{x}{2} \right\rfloor))$ $dist(G_0^{n-1}(y), G_0^{n-2}(\left\lfloor \frac{x}{2} \right\rfloor)) + dist(G_0^{n-1}(x)\overline{G_0^{n-1}(x)}, 0^2 \leq 2$. We further consider the edges linking any node (l, y, x) to its adjacent nodes $(l, y, x+1)$ and $(l, y+1, x)$, $0 \leq x, y \leq 2^l - 1$. When $l = n-1$, it is easy to check that $dist(f_1(n-1, y, x), f_1(n-1, y, x+1)) \leq 2$ and $dist(f_1(n-1, y, x), f_1(n-1, y+1, x)) = 1$. The dilations for the case $0 \leq l < n-1$ are given by $dist(f_1(l, y, x), f_1(l, y, x+1)) = dist(G_0^l(x), G_0^l(x+1)) + dist(G^l(x), G^l(x+1)) \leq 2$ and $dist(f_1(l, y, x), f_1(l, y{+}1, x)) = dist(G^l(y), G^l(y+1)) = 1$. We complete the proof. □

Before presenting the following lemma, we will first define the full-free node in a hypercube. A node in a hypercube is called a full-free node if no pyramid node is embedded into that node and each edge incident to the full-free node is not a part of the path emulating any pyramid edge.

Lemma 3. Using the embedding function f_1, embedding P_n into H_{2n-1} has congestion 2; there exists a full-free node adjacent to the node emulating the apex in P_n along the most significant dimension.

Proof: The congestion 2 is shown by induction on the height of P_n. For $n = 3$, the lemma is true as shown in Fig. 3. There exists a full-free node (11, 000) in H_5 adjacent to node (01, 000) emulating the apex in P_3 along the most significant dimension. For some n, assume that the above lemma is true.

Changing n to $n + 1$, by f_1, the four subpyramid $\overline{P}_n$'s with apexes (1, 0, 0), (1, 0, 1), (1, 1, 0), and (1, 1, 1) are embedded into four subcubes $(0\{0, 1\}^{n-2}, 0\{0, 1\}^{n-1})$, $(0\{0, 1\}^{n-2}, 1\{0, 1\}^{n-1})$, $(1\{0, 1\}^{n-2}, 0\{0, 1\}^{n-1})$, and $(1\{0, 1\}^{n-2}, 1\{0, 1\}^{n-1})$, respectively. According to the hypothesis, embedding each $\overline{P}_n$ into a $(2n - 1)$-dimensional subcube has congestion 2. Consider the edges linking the node-pair $(l, y, 2^{l-1} - 1)$ and $(l, y, 2^{l-1})$, and linking the node-pair $(l, 2^{l-1} - 1, x)$ and $(l, 2^{l-1}, x)$, $2 \le l \le n - 1$. Two nodes in one node--pair belong to the different $\overline{P}_n$ each other. Because $dist(f_1(l, y, 2^{l-1}-1), f_1(l, y, 2^{l-1})) = dist(G_0^l(2^{l-1}-1), G_0^l(2^{l-1})) + dist(G^l(x), G^l(x+1)) =$ 0+1=1, the communication paths emulating the edges do not increase the congestion. Therefore, we will only concentrate on edges linking the apex to its four children and linking any two adjacent nodes at level 1 in P_{n+1}.

By f_1, the five nodes (0, 0, 0), (1, 0, 0), (1, 0, 1), (1, 1, 0), and (1, 1, 1) in P_{n+1} are embedded into $(01^{n-1}, 0^{n+1})$, $(001^{n-2}, 0^{n+1})$, $(011^{n-2}, 10^n)$, $(101^{n-2}, 0^{n+1})$, and $(111^{n-2}, 10^n)$, respectively. Following the hypothesis, there exist four full-free nodes $(011^{n-2}, 0^{n+1})$, $(001^{n-2}, 10^n)$, $(111^{n-2}, 0^{n+1})$, and $(101^{n-2}, 10^n)$ when four $\overline{P}_n$'s in P_{n+1} are embedded into four $(2n - 1)$-dimensional subcubes in H_{2n+1}. After embedding P_{n+1} into H_{2n+1}, the node $(011^{n-2}, 0^{n+1})$ emulates the apex in P_{n+1}. The communication pattern for these five hypercube nodes emulating the corresponding five pyramid nodes is shown in Fig. 4, where the lines denote the paths emulating the edges linking two adjacent nodes at level 1 and linking the apex and its children. Obviously, the congestion is also 2 for these eight communication paths.

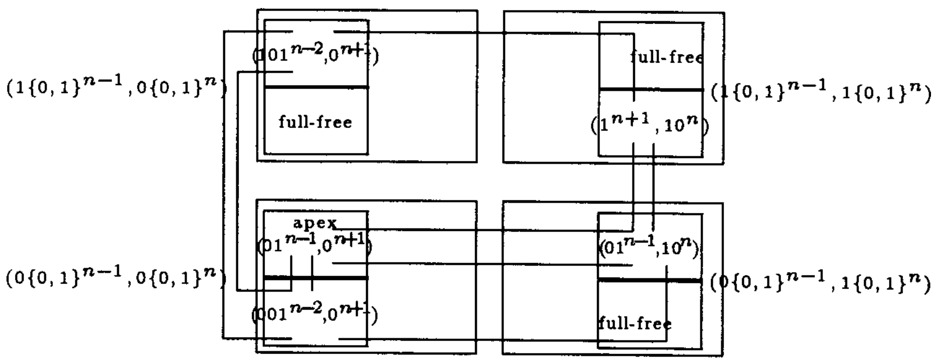

Fig. 4. Communication pattern for the five hypercube nodes emulating the pyramid nodes at the top two levels.

In addition, the full-free node $(111^{n-2}, 0^{n+1})$ is adjacent to the node $(011^{n-2}, 0^{n+1})$ emulating the apex in P_{n+1} along the most significant dimension. We complete the proof.

□

It is easy to verify that the expansion is $2^{2n-1}/((4^n - 1)/3) \approx 3/2$. In addition, O($n$) time is sufficient to translate an n-bit binary string b for $0 \le b \le 2^n - 1$ (binary-reflected Gray code $G^n(b')$) to the corresponding binary-reflected Gray code $G^n(b)$ (n-bit binary string $b'_n b'_{n-1} \cdots b'_2 b'_1 = b'$ for $b'_i \in \{0, 1\}$ and $1 \le i \le n$) [7]. By Lemmas 1, 2, and 3, we have the following result.

Theorem 4. With dilation 2, congestion 2, load 1, and expansion 3/2, P_n can be embedded into H_{2n-1} in O(n) time.

The advantage of the mapping function f_1 is that it can be easily computed due to its closed form representation. From the description of the proposed embedding method, it is important that any node with address (l, y, x) in P_n can be mapped into the corresponding node in H_{2n-1}. That is, our embedding method can be performed in a parallel manner. In contrast, the two embedding methods in [6] used to map P_n into H_{2n-1} are carried out in a recursive manner. In their first embedding method using the bottom-up approach, Lai and White first map P_2 into H_3; then they map one apex and four P_2's into H_5, and so on. In general, after mapping P_k into H_{2k-1}, they further map one apex and four P_k's into H_{2k+1}. The second embedding method presented in [6] uses the recursive top-down approach and is somewhat complicated, so we omit a description due to space limitations. The interested readers are referred to [6].

To summarize, a performance comparison among the previous results [3, 5, 6] and ours for embedding P_n into H_{2n-1} is shown in Table 1.

Table 1. Performance comparison among the five methods for embedding P_n into H_{2n-1}.

method	dilation	expansion	congestion
[6]	3	3/2	2
[6]	2	3/2	3
[5]	2	3/2	2
[3]	2	3/2	2
this paper	2	3/2	2

4. EMBEDDING P_n INTO H_k FOR $0 \le k \le n - 1$

This section presents an embedding function used to map P_n into H_k, $0 \le k < 2n - 1$, with load balancing. If k is odd (even), then we let $k = 2t - 1$ ($k = 2t$), where t is a positive integer. Extending f_1, we will first present an embedding function used to map P_n into H_k, $0 \le k = 2t - 1 < 2n - 1$, with load balancing. The embedding function for the case $k = 2t$ will be discussed later. For clarity, suppose P_n is partitioned into two parts: the top subpyramid $\hat{P}_t$ with t levels and 4^t attached subpyramids $\overline{P}_{n-t}$'s, each with $n - t$ levels. By Theorem 4, the top subpyramid $\hat{P}_t$ can be embedded into H_{2t-1}. The two attached subpyramids $\overline{P}_{n-t}$'s can be thought of as a supernode such that these $\overline{P}_{n-t}$'s form a $2^{t-1} \times 2^t$ mesh which can be embedded into H_{2t-1} with dilation 1 [8, 9]. Based on the above description, the new embedding function f_2 used to map P_n into H_k, $0 \le k = 2t - 1 \le 2n - 2$, is defined as

$$f_2(l,\ y,\ x) = \begin{cases} (G^l(y)G_0^l(x)1^{t-l-2}, G^l(x)0^{t-l}), & 0 \le l < t-1 \\ (G^{t-1}(y), G^{t-1}(x)\overline{G_0^{t-1}(x)}), & l = t-1 \\ (G^{t-1}(\left\lfloor \frac{y}{2^{l-t+1}} \right\rfloor), G^t(\left\lfloor \frac{x}{2^{l-t}} \right\rfloor)), & t \le l \le n-1. \end{cases}$$

Using f_2 with $n = 3$, $k = 3$, and $t = 2$, embedding P_3 into H_3 is shown in Fig. 5. The lines in H_3 denote with paths emulating some edges in P_3, with each edge linking two pyramid nodes which are embedded into two different nodes in H_3. The apex node (0, 0, 0) and two base nodes (2, 0, 0) and (2, 1, 0) in P_3, for example, are embedded into the node 000 (=(0, 00)) in H_3. The edge linking node 000 and 001 in H_3, denoted by <000, 001>, is shared by 6 paths emulating the 6 edges <(2, 0, 0), (2, 0, 1)>, <(2, 1, 0), (2, 1, 1)>, <(1, 0, 0), (2, 1, 0)>, <(1, 0, 0), (2, 0, 0)>, <(0, 0, 0), (1, 0, 0)>, and <(0, 0, 0), (1, 1, 0)> in P_3.

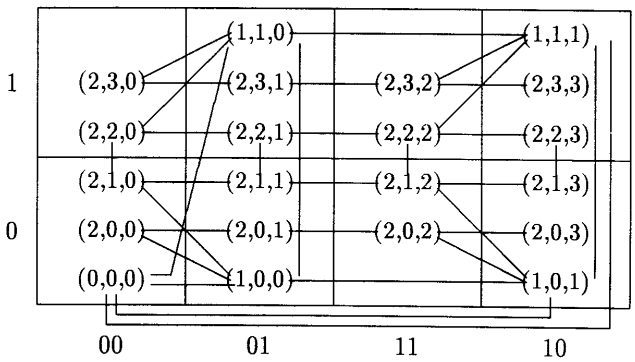

Fig. 5. Embedding P_3 into H_3.

Based on f_2, we have the following theorem.

Theorem 5. With dilation 2, congestion $2^{n-t+1} + 2$, and load $\lceil 2^{2n-k} / 3 \rceil$, P_n can be embedded into H_k, $0 \le k = 2t - 1 \le 2n - 2$.

Proof: There are three same cases to be considered: embedding $\hat{P}_t$ into H_{2t-1}, embedding 4^t attached $\overline{P}_{n-t}$'s into H_{2t-1}, and communicating between the bottom nodes of $\hat{P}_t$ and the apexes of these $\overline{P}_{n-t}$'s. In the first case, by Theorem 4, $\hat{P}_t$ can be embedded into H_{2t-1} with dilation 2, congestion 2, load 1, and expansion 3/2. In the second case, two attached subpyramids $\overline{P}_{n-t}$'s are thought of as a supernode such that these $\overline{P}_{n-t}$'s form a $2^{t-1} \times 2^t$ mesh which can be embedded into H_{2t-1} with dilation 1 [8, 9]. It is easy to verify that this case has dilation 1, congestion $2^{n-t+1} - 2$, and load $\lceil (2 \cdot 4^{n-t} - 2) / 3 \rceil$. For the third case, by f_2, it is easy to see that all the nodes at the bottom level of $\hat{P}_t$ are embedded into the nodes with labeling $\{0, 1\}^{2t-3}01$ or $\{0, 1\}2^{2t-3}10$. As mentioned in the second case, a $2^{t-1} \times 2^t$ mesh is embedded into H_{2t-1}. Hence, any bottom node in $\hat{P}_t$ and its two children are embedded into the same hypercube node, say $b_{2t-2}b_{2t-3}\cdots b_2 01$ (or $b_{2t-2}b_{2t-3}\cdots b_2 10$), and the other two children are embedded into its adjacent node $b_{2t-2}b_{2t-3}\cdots b_2 00$ (or $b_{2t-2}b_{2t-3}\cdots b_2 11$), $b_i \in \{0, 1\}$ for $2 \le i \le 2t - 2$ (see Fig. 3). Thus, it has dilation 1 and congestion 2. As a result, P_n can be embedded into H_k with dilation 2 (= max{2, 1, 1}), congestion $2^{n-t+1} + 2$ (= $2 + 2^{n-t+1} - 2 + 2$), and load $\lceil 2^{2n-k}/3 \rceil$(=$(2 \cdot 4^{n-t} + 1)/3 = (2 \cdot 4^{n-t} - 2)/3 + 1$). □

Consider the other case, $k = 2t$. The new embedding function f_3 can be obtained by extending f_2 slightly, and it is given by

$$f_3(l, y, x) = \begin{cases} (G^l(y)G_0^l(x)1^{t-l-2}0, \overline{G^l(x)}0^{t-l}), & 0 \le l < t-1 \\ (G^{t-1}(y)0, G^{t-1}(x)\overline{G_0^{t-1}(x)}), & l = t-1 \\ (G^t(\lfloor \frac{y}{2^{l-t}} \rfloor), G^t(\lfloor \frac{x}{2^{l-t}} \rfloor)), & t \le l \le n-1. \end{cases}$$

The three cases to be considered are the same as those in Theorem 5. In the first case, by Theorem 4, $\hat{P}_t$ is embedded into one half of H_{2t}, $(\{0, 1\}^{n-1}0, \{0, 1\}^n)$. In the second case, each attached subpyramid $\overline{P}_{n-t}$ can be thought of as a supernode such that these $\overline{P}_{n-t}$'s form a $2^t \times 2^t$ mesh which can be embedded into H_{2t} with dilation 1, congestion $2^{n-t} - 1$, and load $\lceil 4^{n-t} - 1)/3 \rceil$. In the third case, each bottom node of $\hat{P}_t$ say $(t-1, \lfloor \frac{y}{2} \rfloor, \lfloor \frac{x}{2} \rfloor)$, communicates with its four children, apexes of $\overline{P}_{n-t}$'s, say (t, y, x), $(t, y, x + 1)$, $(t, y + 1, x)$, and $(t, y + 1, x + 1)$. These five nodes in P_n are embedded into $(G^{t-1}(\lfloor \frac{y}{2} \rfloor)0, G^{t-1}(\lfloor \frac{x}{2} \rfloor)\overline{G^{t-1}(\lfloor \frac{x}{2} \rfloor)})$, and $(G^t(y), G^t(x))$, $(G^t(y), \overline{G^t(x+1)})$, $(G^t(y+1), G^t(x))$, and $(G^t(y+1), G^t(x+1))$. The node $(G^{t-1}(\lfloor \frac{y}{2} \rfloor)0, G^{t-1}(\lfloor \frac{x}{2} \rfloor)\overline{G^{t-1}(\lfloor \frac{x}{2} \rfloor)})$ and one of the other four mapped nodes are the same. These four mapped hypercube nodes can be thought as of a ring with length 4. For instance, the five nodes (2, 1, 0), (3, 2, 0), (3, 2, 1), (3, 3, 0), and (3, 3, 1) in P_5 are mapped into the five nodes (010, 001), (011, 000), (011, 001), (010, 000), and (010, 001) in H_6, respectively. It is easy to see that both nodes (2, 1, 0) and (3, 3, 1) in P_5 are mapped into the same node (010, 001) in H_6. Therefore, this case has dilation 2 and congestion 2. Consequently, the embedding has dilation 2 (= max{2, 1, 2}), congestion $2^{n-t} + 3$ (= $2 + 2^{n-t} - 1 + 2$), and load $\lceil 2^{2n-k}/3 \rceil$ $(= (4^{n-t} + 2)/3 = (4^{n-t} - 1)/3 + 1)$. We have the following corollary.

Corollary 6. With dilation 2, congestion $2^{n-t} + 3$, and load $\lceil 2^{2n-k}/3 \rceil$, P_n can be embedded into H_k, $0 \le k = 2t \le 2n - 2$.

Based on f_2 and f_3, this embedding can be accomplished in a parallel manner. Because O(k) time is sufficient to translate a k-bit binary string b for $0 \le b \le 2^k - 1$(binary-reflected Gray code $G^k(b')$) into the corresponding binary-reflected Gray code $G^k(b)$ (k-bit binary string $b'_k b'_{k-1} \cdots b'_2 b'_1 = b'$ for $b'_i \in \{0, 1\}$ and $1 \le i \le k$) [7], we have the following result.

Corollary 7. Embedding P_n into H_k can be accomplished using the above parallel algorithm in O(k) time.

5. DISCUSSION AND REMARKS

Our major contribution in this short paper has been to present a parallel embedding algorithm which can be used to map large pyramids into smaller hypercubes with load balancing. With dilation 2, congestion $2^{n-t} + 3$ (or $2^{n-t+1} + 2$), and load $\lceil 2^{2n-k} / 3 \rceil$ when $k = 2t$ (or $k = 2t - 1$), our method can embed P_n into H_k in $O(k)$ time.

ACKNOWLEDGMENTS

The authors would like to thank the four anonymous reviewers for their constructive comments for improving the quality and presentation of this new version.

REFERENCES

1. S. Tanimoto and T. J. Ligoki, "A prototype pyramid machine for hierarchical cellular logic," *Parallel Computer Vision*, Orlando, FL, Academic Press, 1987, pp. 43-83.
2. R. Miller and Q. F. Stout, "Data movement techniques for the pyramid computer," *SIAM Journal on Computing*, Vol. 16, No. 1, 1987, pp. 38-60.
3. Q. F. Stout, "Hypercubes and pyramids," *Pyramidal Systems for Computer Vision*, V. Cantoni and S. Levialdi, eds., Springer, 1986, pp. 75-89.
4. F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann Pub., CA., Chap. 3, 1992.
5. B. Monien and H. Sudborough, "Embedding one interconnection network in another," *Computing* [*Supplementum*], No. 7, 1990, pp. 257-282.
6. T. H. Lai and W. White, "Mapping pyramid algorithms into hypercubes," *Journal of Parallel and Distributed Computing*, Vol. 9, No. 1, 1990, pp. 42-54.
7. F. Annexstein, "Parallel implementations of graph embeddings," *Lecture Notes in Computer Science 678*, 1993, pp. 207-217.
8. S. L. Johnsson, "Communication efficient basic linear algebra computations," *Journal of Parallel and Distributed Computing*, Vol. 4, No. 2, 1987, pp. 133-172.
9. Y. Saad and M. H. Schultz, "Topological properties of hypercube," *IEEE Transactions on Computers*, Vol. 37, No. 7, 1988, pp. 867-872.

Yu-Wei Chen (**陳育威**) received the B.S. and Ph.D. degrees from the Department of Information Management of National Taiwan University of Science and Technology, R.O.C., in 1993 and 1999, respectively. In 1994, he began work toward the Ph.D. degree after taking first year courses for the M.S. degree. He is now an assistant professor at Tamsui Oxford University College. His research interests include parallel and distributed computing, fault-tolerant computing, and networks.

Kuo-Liang Chung (鍾國亮) received the B.S., M.S., and Ph.D. degrees in Computer Science and Information Engineering from National Taiwan University, R.O.C. He is a professor in the Department of Information Management and the Institute of Information Engineering of National Taiwan University of Science and Technology. His current research interests include image processing, compression, computer graphics, computational geometry, and theoretical computer science. He is a member of ACM and IEEE.