

Parallel Adaptive Estimation of Hip Range of Motion for Total Hip Replacement Surgery*

Yasuhiro KAWASAKI^{†a)}, Student Member, Fumihiko INO^{†b)}, Yoshinobu SATO^{††}, Shinichi TAMURA^{††}, and Kenichi HAGIHARA[†], Members

SUMMARY This paper presents the design and implementation of a hip range of motion (ROM) estimation method that is capable of fine-grained estimation during total hip replacement (THR) surgery. Our method is based on two acceleration strategies: (1) adaptive mesh refinement (AMR) for complexity reduction and (2) parallelization for further acceleration. On the assumption that the hip ROM is a single closed region, the AMR strategy reduces the complexity for $N \times N \times N$ stance configurations from $O(N^3)$ to $O(N^D)$, where $2 \leq D \leq 3$ and D is a data-dependent value that can be approximated by 2 in most cases. The parallelization strategy employs the master-worker paradigm with multiple task queues, reducing synchronization between processors with load balancing. The experimental results indicate that the implementation on a cluster of 64 PCs completes estimation of $360 \times 360 \times 180$ stance configurations in 20 seconds, playing a key role in selecting and aligning the optimal combination of artificial joint components during THR surgery.

key words: range of motion estimation, adaptive mesh refinement, cluster computing, medical image processing, computer assisted surgery

1. Introduction

Total hip replacement (THR) [1], [2] is a surgical procedure that relieves patients of hip pain and removes their difficulty in walking by replacing the hip joint with an artificial joint. The key issue in this surgery is to select and align the optimal combination of artificial joint components: the cup, head, neck, and stem components as illustrated in Fig. 1 (a). These optimal selection and alignment are important for both the surgeon and patient because either inappropriate or malpositioned components increase the risk of clinical problems such as dislocation, wear, and loosening [2], [3].

In order to assist the surgeon in finding the best combination and placement of joint components, range of motion (ROM) estimation systems [3], [4] have been developed in the past. Earlier systems are useful in developing preoperative surgical plans because they present the limitation of hip movement on three-dimensional (3-D) polygonal surface models reconstructed from patients' computed tomog-

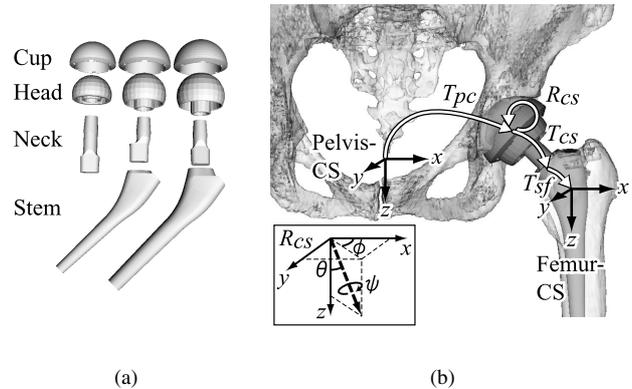


Fig. 1 (a) Components of artificial joints and (b) representation of hip joint motion.

raphy (CT) images. However, the preoperative plans may need to be changed by unexpected conditions that reveal during surgery. For example, if the bone tissue around the preoperatively planned position is known to be fragile, another position must be selected according to intraoperative circumstances. Thus, intraoperative estimation is essential to overcome such unexpected conditions.

One issue to develop intraoperative estimation systems is a large amount of computation due to collision detections (CDs) required for ROM estimation. For example, an earlier system [5] on a Sun Ultra 30 running at 300 MHz takes 0.05 seconds to detect a collision for a stance configuration, so that it takes approximately 13 days to compute a 3-D ROM with $360 \times 180 \times 360 = 23,328,000$ configurations: 360° , 180° , and 360° for yaw (ϕ), pitch (θ), and roll (ψ) angles, respectively, as illustrated in Fig. 1 (b). Therefore, to complete an estimation job during surgery, earlier systems need to degrade the quality of estimations by limiting the area of estimation [3] or by reducing the number of stance configurations [6].

In contrast to the coarse-grained estimation mentioned above, the key contribution of this paper is to provide fine-grained estimation during surgery. To achieve this, we have developed a fast estimation method based on two key strategies: (1) adaptive mesh refinement (AMR) for complexity reduction and (2) parallelization for further acceleration. The analytical results show that the AMR strategy, which assumes that the hip ROM is a single closed region, reduces the time complexity for most clinical datasets. Furthermore,

Manuscript received February 28, 2006.

Manuscript revised June 30, 2006.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Toyonaka-shi, 560-8531 Japan.

^{††}The authors are with the Graduate School of Medicine, Osaka University, Suita-shi, 565-0871 Japan.

*A preliminary version of this paper was presented at the 7th Int'l Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI 2004).

a) E-mail: y-kawask@ist.osaka-u.ac.jp

b) E-mail: ino@ist.osaka-u.ac.jp

DOI: 10.1093/ietisy/e90-d.1.30

the experimental results indicate that the combination of the two strategies is 11 times faster than the prior parallelization strategy [6] that samples stance configurations by a uniform mesh instead of an adaptive mesh.

The remainder of the paper is organized as follows. Section 2 gives an overview of ROM estimation and CD algorithms. Section 3 presents the design and implementation of our estimation method. Section 4 gives analytical results using fractal theory. Section 5 shows some experimental results obtained on a cluster system [7]. Section 6 introduces some related works. Finally, Sect. 7 concludes the paper.

2. Background

2.1 Range of Motion (ROM) Estimation

To describe the details of ROM estimation, we first show a brief representation of hip joint motion described in [3].

Let M_{pf} denote the transformation from the pelvis coordinate system (pelvis-CS) to the femur coordinate system (femur-CS). As illustrated in Fig. 1 (b), the hip joint motion is given by:

$$M_{pf} = T_{pc}T_{cs}R_{cs}T_{sf}, \quad (1)$$

where T_{pc} is a 4×4 transformation matrix representing the orientation of the cup in the pelvis-CS, T_{cs} is a fixed transformation matrix determined by the selected head and neck components, R_{cs} is a variable transformation matrix constrained to the rotational motion, and T_{sf} is a transformation matrix representing the reverse orientation of the stem in the femur-CS. Both T_{pc} and T_{sf} are determined by one of the following two methodologies. For preoperative surgical planning, the medical doctor determines them by experience using visual guides, as shown in Fig. 1 (b). On the other hand, for intraoperative assistance, optical 3-D position sensors give the actual values of T_{pc} and T_{sf} by measuring implanted components. Thus, intraoperative estimation based on measured T_{pc} and T_{sf} is required to obtain exact ROMs.

Given T_{pc} , T_{cs} , and T_{sf} , the safe ROM is defined as a set of rotation transformation matrices, \mathcal{S} , such that for all $R_{cs} \in \mathcal{S}$, R_{cs} avoids any implant-implant, bone-implant, and bone-bone impingements. Thus, the problem of ROM estimation can be resolved by checking collisions among these objects with varying stance configurations. Figure 2 shows an example of the safe ROM presenting two features as follows:

- a closed line shapes the boundary between inside and outside the ROM;
- the safe ROM does not always include the origin.

Therefore, ROM estimation is a search problem that locates the boundary of a closed region in 3-D space. Since R_{cs} is defined in 3-D space, ROM estimation is a computationally intensive application. In the following, we represent R_{cs} by the Euler angles, (ϕ, θ, ψ) ($0^\circ \leq \phi < 360^\circ$, $0^\circ \leq \theta < 180^\circ$, $-180^\circ \leq \psi < 180^\circ$), as shown in Fig. 1 (b).

Figure 3 shows how prior methods [3], [4], [6] compute

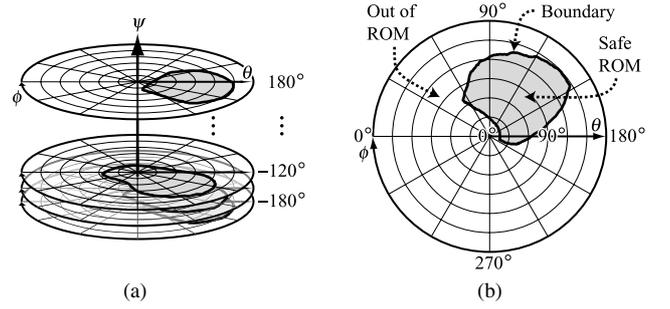


Fig. 2 (a) A 3-D ROM and (b) its slice, where $\psi = 30^\circ$, given in polar coordinates. The entire region in the slice is separated into two pieces by a closed line. An enclosed region represents the safe ROM while the outside region represents the out of the ROM.

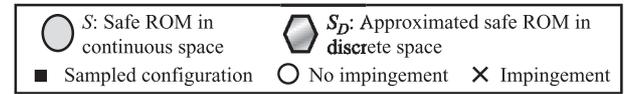
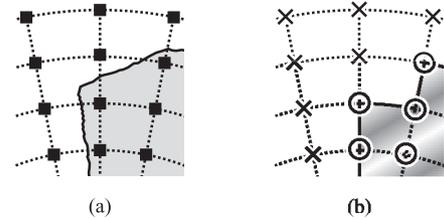


Fig. 3 (a) The ROM \mathcal{S} in continuous space is estimated using a uniform mesh of sampled stance configurations in discrete space. (b) Then, it is approximated by a discrete region \mathcal{S}_D that covers all collision-free configurations.

\mathcal{S} . To estimate \mathcal{S} , these methods sample every stance configuration (ϕ, θ, ψ) at uniform intervals, and then investigate each configuration whether it has impingements. After this, they approximate \mathcal{S} by a discrete region \mathcal{S}_D that covers all collision-free stance configurations. Therefore, the amount of computation increases with the number of stance configurations. Furthermore, it depends on the complexity of the CD algorithm employed for each stance configuration.

2.2 Collision Detection (CD)

The basic problem of CD between two objects is to determine whether or not they geometrically overlap each other. For polygonal surface models, where a set of planar polygons constructs an object, this problem can be resolved into the detection of overlaps between polygons. A naive approach is to test all pairs of polygons. However, this is not efficient for our case, because our target objects are precisely modeled using many polygons. Therefore, some methods are required to approximate objects by a simple shape that circumscribes an object.

Thus, efficient CD algorithms usually approximate an object by a hierarchy of bounding volumes (BVs) such as a tree of oriented bounding boxes (OBBs) [8], that of spheres [9], and that of discrete orientation polytopes (k -DOPs) [10]. Such algorithms perform CDs between BVs instead of ob-

Algorithm AMR_based_ROM_estimationInputs: 3-D surface models and the finest level F Output: \mathcal{S}_D : Discrete ROM

```

1: Initialize level  $k := 1$ ;
2: Initialize set  $P_0 := \emptyset$ ; // a set of lattice points
3: Generate a uniform mesh  $G_1$  with the coarsest interval  $i_1$ ;
4: while  $k \leq F$  do
5:   Initialize set  $P_k := \emptyset$ ;
6:   for each lattice point  $(\phi, \theta, \psi)$  of mesh  $G_k$  do
7:     Check collisions for stance configuration  $(\phi, \theta, \psi)$ 
      (currently by the V-COLLIDE library);
8:     if  $(\phi, \theta, \psi)$  is collision-free then
9:       Add configuration  $(\phi, \theta, \psi)$  to set  $P_k$ ;
10:    end
11:    $P_k := P_k \cup P_{k-1}$ ; //  $P_k$  contains all collision-free points
12:   if  $k < F$  then
13:     for each lattice point  $(\phi, \theta, \psi)$  of mesh  $G_k$  do
14:       Determine the status of  $(\phi, \theta, \psi)$  according to Eq. (2);
15:     end
16:     Generate mesh  $G_{k+1}$  with refined interval  $i_{k+1} := i_k/2$ 
      such that  $G_{k+1}$  covers all of the active points in  $G_k$ ;
17:   end
18:   Update level  $k := k + 1$ ;
19: end
20: Set  $\mathcal{S}_D$  such that  $\mathcal{S}_D$  is a single enclosed region covered by all
    collision-free points  $P_F$ ;

```

Fig. 4 ROM estimation algorithm based on adaptive mesh refinement strategy. P_k represents a set of collision-free points obtained after the k -th level estimation.

jects in order to immediately stop the test if the objects are not apparently located in close proximity. Otherwise, namely if the BVs have overlaps due to closely located objects, the algorithms move up the hierarchy of BVs to perform more exact CDs using tighter approximation. Thus, efficient CD algorithms prune overlap tests for some pairs of polygons by using geometric approximation of objects.

Prior ROM estimation systems [3]–[6] use the V-COLLIDE library [11] for CDs, which is based on a hierarchy of OBBs. This hierarchy provides efficient pruning for objects composed of many polygons, such as bone surfaces, because it realizes tight approximation of such objects. Our estimation system also uses the V-COLLIDE library for CDs.

3. Fast ROM Estimation

This section describes the AMR strategy and the parallelization strategy used in our method. These key strategies are layered on top of the V-COLLIDE library. The AMR strategy aims at minimizing calls of V-COLLIDE functions while the parallelization strategy focuses on increasing the throughput of our cluster system.

3.1 Adaptive Mesh Refinement (AMR)

The key idea of AMR is to investigate in detail the stance configurations only close to the boundary of the ROM. To realize such non-uniform refinement, our estimation algorithm employs two techniques as follows (see also Fig. 4).

Hierarchical structure of meshes. The AMR strategy em-

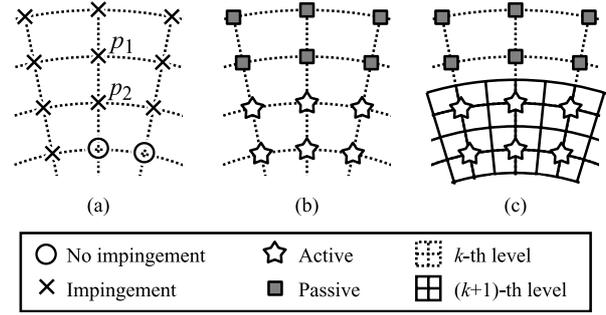


Fig. 5 Process of adaptive mesh refinement. (a) For all configurations sampled at the k -th level, the algorithm checks collisions. (b) Then, it determines the status of each stance configuration according to Eq. (2). In this example, configuration p_1 is marked as “passive” because p_1 and its surrounding configurations have the same CD result (impingements, in this case). In contrast, p_2 is an active configuration because two of its neighbors have no impingement while p_2 has impingements. (c) The $(k + 1)$ -th mesh is generated only around active configurations.

plies a hierarchical structure of meshes to sample stance configurations. A stance configuration here corresponds to a lattice point of a mesh. At the beginning of the estimation, the algorithm generates the initial uniform mesh G_1 with the coarsest sampling intervals i_1 . After the k -th level estimation, where $k \geq 1$, it generates the refined mesh G_{k+1} with half intervals $i_{k+1} = i_k/2$ only around configurations potentially close to the boundary of the ROM. Thus, refinements around the remaining configurations are omitted in order to reduce the amount of computation. These refinements are repeated until reaching the finest level F .

Status control of each stance configuration. Figure 5 shows the process of selecting configurations close to the boundary. Our algorithm associates each configuration with a status, $C_{\phi, \theta, \psi} \in \{\text{active, passive}\}$. “Active” here means that the configuration is potentially close to the boundary while “passive” means that the configuration is apparently far from the boundary. These decisions are based on the assumption that the hip ROM is a single closed region. Under this assumption, a stance configuration surrounded by neighbors with impingements (or no impingements) also has impingements (or no impingements, respectively). Therefore, such a surrounded configuration is marked as “passive” in order to avoid further mesh refinement around the configuration. In summary, the status of each stance configuration is given by:

$$C_{\phi, \theta, \psi} = \begin{cases} \text{passive,} & \text{if } \forall p, q, r \in \{-i_k, 0, i_k\} \\ & [(\phi+p, \theta+q, \psi+r) \in P_k] \\ & \vee \forall p, q, r \in \{-i_k, 0, i_k\} \\ & [(\phi+p, \theta+q, \psi+r) \notin P_k] \\ \text{active,} & \text{otherwise.} \end{cases} \quad (2)$$

where P_k represents a set of collision-free configurations computed at the k -th level estimation.

If the k -th level mesh does not contain all of the configurations $(\phi+p, \theta+q, \psi+r)$ in Eq. (2), the status $C_{\phi, \theta, \psi}$ is decided according to the match or mismatch of the CD results only for sampled configurations.

Note that our adaptive algorithm does not always give the same results as non-adaptive algorithms. This difference is due to the assumption mentioned above. For example, if the initial sampling intervals are too large, our algorithm may fail to detect a small ROM hidden entirely in a large cell of the coarse mesh. However, we think that we can easily avoid such inappropriate initial intervals, because clinical datasets allow us to roughly estimate the regional size of the ROM according to clinical statistics.

Another solution to detect such small cells is to employ a classical trick known in the field of computational geometry [12]. This trick uses dilated objects to check possible collisions for continuous range of configurations. A dilated object is a trajectory of the original objects rotating from a sampled configuration to a next sampled configuration. If the dilated object is free from collisions with the other object, the trick ensures that no impingement occurs at any configurations between two sampled configurations. Thus, the trick reveals smaller ROMs, allowing our adaptive algorithm to produce the same results as non-adaptive algorithms.

In addition to the complexity reduction, our adaptive algorithm has another advantage compared with non-adaptive algorithms. Since our algorithm refines the mesh as it moves up the hierarchy, it enables progressive visualization of the safe ROM. Therefore, an outline of the safe ROM is roughly visible in the early phase of estimation, allowing surgeons to immediately terminate the ongoing estimation job, if the outline is known to be apparently a unoptimal result.

3.2 Parallelization

In our algorithm, a CD task for stance configuration (ϕ', θ', ψ') is dependent on that for another configuration (ϕ, θ, ψ) if and only if status $C_{\phi, \theta, \psi}$ determines whether to sample (ϕ', θ', ψ') or not at mesh refinement. In other words, any two configurations without causal relations between them can be processed in parallel. Therefore, our strategy exploits task parallelism in ROM estimation. In the following discussion, a task corresponds to CDs for a set of configurations in the same level or in different levels but without causal relations among the configurations.

In order to exploit this parallelism, our strategy employs the master-worker (MW) paradigm, where computing nodes are classified into two groups: a master and the remaining workers. Figure 6 shows an overview of our parallelization strategy. While the workers check collisions for tasks assigned from the master, the master manages estimation results, namely the safe ROM, and updates the status of each configuration. According to this status, the master determines which region need to be further refined, and then samples stance configurations for the next level. These newly sampled configurations are enqueued as tasks to a queue prepared for each estimation level. Tasks are dequeued when assigning them to workers. Since idle workers are selected for this assignment, the MW paradigm is capa-

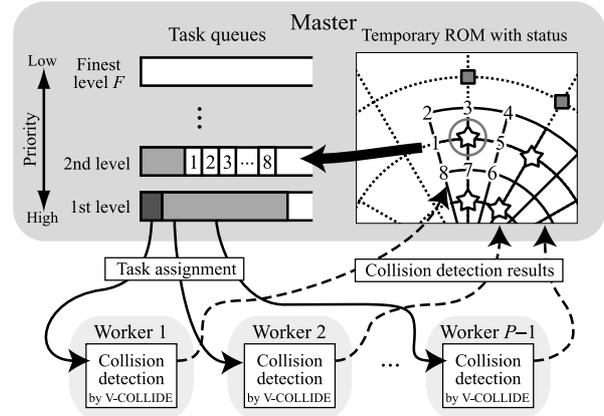


Fig. 6 Adaptive ROM estimation based on the master-worker paradigm. Given P computing nodes, $P-1$ workers check collisions for tasks assigned from a master while the master manages estimation results and task queues for each estimation level. The master is also responsible for mesh refinement. In this example, eight configurations around an active configuration are newly sampled and enqueued to the corresponding queue.

ble of dynamic load balancing among workers.

Note here that our parallelization strategy prevents synchronization among workers when advancing the estimation level. If we implement the MW paradigm in a naive way in which the master needs all results of the current level before advancing to the next level, idle workers have to wait for the master to compute the status of returned configurations and to sample new configurations for the next level (see lines 11–17 in Fig. 4).

To eliminate this synchronization, our strategy uses an asynchronous mechanism in which the master determines the status and samples stance configurations immediately after receiving CD results from the workers. This asynchronous behavior achieves higher throughput because it minimizes waiting time at the workers. However, the master must simultaneously manage tasks sampled from different levels. Such tasks should be processed from the coarse-grained level because tasks at coarser levels can generate many independent tasks after refinement. Therefore, as shown in Fig. 6, the master has multiple queues with priorities to enable this coarse-to-fine processing.

The load-balancing nature of the MW paradigm is important to achieve higher speedup, because different CD tasks have different computational loads in our estimation algorithm. This difference is due to the V-COLLIDE library, which employs BVs to dynamically prune overlap tests, depending on the position and orientation of objects. Actually, the execution time per configuration ranges from 0.1 to 10 milliseconds.

Generally, we should carefully select the grain size of tasks in order to achieve higher speedup for MW applications. If tasks are excessively fine-grained, the workers always wait for the master, which suffers in frequent task assignments. In contrast, if tasks are excessively coarse-grained, the MW paradigm loses the load-balancing effects. Thus, an appropriate grain size is necessary to obtain the

load-balancing effects for higher speedup.

In our current system, the grain size of a task, namely the number of configurations that compose a task, is experimentally determined according to the entire execution time. We currently use the same grain size for all estimation levels. Therefore, our grain size will not be optimal for each of estimation levels. However, according to experimental results, we think that the uniform solution is reasonable in terms of efforts to get satisfactory performance.

4. Analytical Results

To theoretically evaluate our adaptive algorithm, we now analyze its time complexity and compare it with a non-adaptive algorithm that employs a uniform mesh. For a measure of the time complexity, we use the number of sampled configurations, L , namely the total number of (V-COLLIDE) function calls required for CDs. We assume that the computational cost of mesh generation is small enough to ignore, as compared with that of CDs. The parallel performance of our method is beyond the scope of this evaluation, and shown in the next section.

The following analysis shows that, given $N \times N \times N$ stance configurations, our algorithm performs $O(N^D)$ CDs instead of $O(N^3)$ CDs, where $2 \leq D \leq 3$ and D is a data-dependent value that can be approximated by 2 in most cases. The key idea in the analysis is to use fractal dimension [13], which characterizes how densely a geometric object occupies the space in which it lies. We use this geometric characteristic to explain how finely the adaptive mesh samples configurations in space, and thus L can be analyzed.

4.1 Complexity Analysis

Let L_k be the number of sampled configurations at the k -th level, where $k \geq 1$. Then, the total number of sampled configurations is given by $L = \sum_{k=1}^F L_k$. The main result of the analysis is as follows.

Theorem 1. Let $\partial\mathcal{S}$ be the boundary of the ROM \mathcal{S} in continuous space. Suppose that \mathcal{S} is a single closed region and $\partial\mathcal{S}$ has fractal dimension D [13]. Then, the adaptive mesh generated by our algorithm has $L = O(N^D)$ lattice points.

To prove Theorem 1, we introduce three lemmas and a definition taken from fractal theory [13].

Lemma 1. Let A_k be the number of active points at the k -th level estimation, where $k \geq 1$. Then, for all $k \geq 1$, the number L_{k+1} of points sampled newly around the active points for the next $(k+1)$ -th level estimation is bounded as follows:

$$(2^3 - 1)A_k \leq L_{k+1} \leq (3^3 - 1)A_k. \quad (3)$$

Proof. Since our algorithm uses 3-D meshes, any active point generates $3^3 - 1$ neighbors after mesh refinement. The worst case occurs when active points are sparsely populated, so that they do not have common neighbors. In this case,

every active point generates $3^3 - 1$ different neighbors after mesh refinement. Therefore, $L_{k+1} \leq (3^3 - 1)A_k$. On the other hand, the lower bound occurs when active points are located contiguously in the mesh, so that some of $3^3 - 1$ neighbors are identical. In this case, every active point has one common neighbor for each of dimensions. Thus, $((3 - 1)^3 - 1)A_k \leq L_{k+1}$. \square

Lemma 2. Let B_k be the number of cubic cells of the mesh G_k that intersect the boundary $\partial\mathcal{S}$. Then, for all $k \geq 1$,

$$A_k \leq 2^3 B_k. \quad (4)$$

Proof. According to Eq. (2), any active point p at the k -th level has at least one neighbor q with a different CD result. That is, p and q (or q and p) are collision-free and collisional points. Therefore, $\partial\mathcal{S}$ exists between them. Furthermore, since p and q are neighboring points, for any active point p , there exists at least one cubic cell that intersects $\partial\mathcal{S}$. The number of such cubic cells is at most 2^3 , because any point of the 3-D mesh is shared with 2^3 cubic cells. \square

Definition 1. Let \mathcal{X} be a non-empty bounded subset of d -dimensional space and let $B_\delta(\mathcal{X})$ be the number of d -dimensional mesh cubes of side length δ that intersect \mathcal{X} . Let $D(\mathcal{X})$ denote the fractal dimension of \mathcal{X} . $D(\mathcal{X})$ is the logarithmic rate at which $B_\delta(\mathcal{X})$ increases as $\delta \rightarrow 0$. Thus, $D(\mathcal{X})$ satisfies

$$B_\delta(\mathcal{X}) \propto \delta^{-D(\mathcal{X})}, \quad (5)$$

where \propto denotes a proportional relation [13].

Lemma 3. If the boundary $\partial\mathcal{S}$ have fractal dimension D , then $B_k \propto i_k^{-D}$, for all $k \geq 1$.

Proof. Due to the assumption that \mathcal{S} is a single closed region, the k -th level mesh G_k includes all cells that intersect the boundary $\partial\mathcal{S}$. Furthermore, G_k consists of cells with the side length i_k , so that B_k corresponds to $B_{i_k}(\partial\mathcal{S})$ in Definition 1. Thus, substituting $\partial\mathcal{S}$ for \mathcal{X} in Definition 1 gives the proof. \square

We now prove Theorem 1.

Proof of Theorem 1. According to Lemma 1, 2, and 3, we have

$$L_k = O(B_{k-1}) = O(i_{k-1}^{-D}), \quad (6)$$

where $k \geq 2$. On the other hand, L_1 is a constant because the algorithm uses a uniform mesh at the first level. Thus, we can claim that

$$\begin{aligned} L &= L_1 + \sum_{k=2}^F L_k \\ &= O(1) + \sum_{k=2}^F O(i_{k-1}^{-D}) \\ &= O(1) + O(i_F^{-D}) = O(N^D). \end{aligned} \quad (7)$$

\square

Next, we analyze the fractal dimension D of the boundary ∂S .

Theorem 2. For any boundary ∂S of three-dimensional ROM S , its fractal dimension D is bounded as follows:

$$2 \leq D \leq 3. \quad (8)$$

To prove Theorem 2, we use the following two properties [13].

Property 1. For two non-empty bounded subsets of d -dimensional space, \mathcal{X} and \mathcal{Y} , if $\mathcal{X} \subset \mathcal{Y}$ then $D(\mathcal{X}) \leq D(\mathcal{Y})$.

Property 2. If \mathcal{X} is a smooth (or continuously differentiable) m -dimensional submanifold (or m -dimensional surface) of d -dimensional space, where $d \geq m$, then $D(\mathcal{X}) = m$.

Proof of Theorem 2. A cube \mathcal{A} larger than S can cover ∂S : $\partial S \subset \mathcal{A}$. Furthermore, ∂S consists of small smooth surfaces \mathcal{B} : $\mathcal{B} \subset \partial S$. On the other hand, Property 2 shows that a cube \mathcal{A} and a smooth surface \mathcal{B} have fractal dimension 3 and 2, respectively: $D(\mathcal{A}) = 3$ and $D(\mathcal{B}) = 2$. Therefore, Property 1 completes the proof. \square

4.2 Discussion for Clinical Datasets

We show that the fractal dimension D of the boundary of the safe ROM can be approximated by 2 for clinical datasets.

As shown in Fig. 7, this is experimentally verified by measuring D for several safe ROMs. The values in Fig. 7 are obtained by applying the box-counting method [13] to each of the estimated safe ROMs. The box-counting method here is a standard method for computing fractal dimensions of geometric shapes. We think that this approximation is applicable to most clinical datasets, because boundaries of hip joint ROMs can be assumed as approximately smooth surfaces due to the following reason.

The boundary ∂S represents the set of the configurations where objects are just touching each other. The objects, the pelvis, femur, and implants in our case, have smooth surfaces. Then, for any fixed yaw angle ϕ , the just-touching configurations (ϕ, θ, ψ) should form a smooth curve in the θ - ψ plane due to the smoothness of surfaces. Similarly, the just-touching configurations should form a smooth curve in the ψ - ϕ and ϕ - θ planes. Therefore, the boundary ∂S should be a smooth surface in configuration space. According to Property 2, such a surface has dimension $D = 2$. Thus, we think that the complexity of our algorithm is $O(N^2)$ for most medical datasets.

5. Experimental Results

In this section, we evaluate the performance of our adaptive method by comparing it with a non-adaptive method [6] on

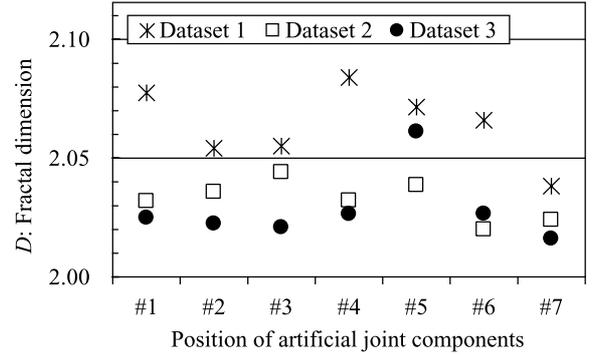


Fig. 7 Fractal dimensions measured for the boundary of the safe ROM. Three different datasets are investigated with seven different positions of artificial joint components.

Table 1 Execution time and its breakdown for fine-grained ROM estimation with $360 \times 180 \times 360$ stance configurations. UM denotes non-adaptive estimation using a uniform mesh.

Breakdown	Sequential		Parallel on 128 CPUs			
			Ethernet		Myrinet	
	UM	AMR	UM	AMR	UM	AMR
V-COLLIDE Initialization	6					
1st level, $i_1 = 16^\circ$	—	6	—	0.1	—	0.1
2nd level, $i_2 = 8^\circ$	—	10	—	0.1	—	0.1
3rd level, $i_3 = 4^\circ$	—	43	—	0.8	—	0.5
4th level, $i_4 = 2^\circ$	—	202	—	3.1	—	2.3
5th level, $i_5 = 1^\circ$	16,318	996	191	16.8	135	10.7
Total time (second)	16,324	1,263	197	26.8	141	19.6

a cluster of PCs. The cluster consists of 64 symmetric multiprocessor (SMP) nodes, each with two Pentium III CPUs running at 1 GHz clock speed. Computing nodes are interconnected by Myrinet [14] and Fast Ethernet, which provide bandwidth of 2 Gb/s and 100 Mb/s, respectively.

We have implemented the method using the C++ language and the MPICH-SCORE library [15], a highly portable, efficient implementation of the Message Passing Interface (MPI) standard [16]. Note here that the MPICH-Score library performs intra-node MPI communication through shared-memory.

The datasets of the pelvis and femur were composed of 116,270 and 30,821 polygons, respectively. See [6] for the detailed explanation on how we generated them. The estimation hierarchy of our adaptive method was composed of five levels ($F = 5$) ranging from $i_1 = 16^\circ$ to $i_5 = 1^\circ$. As well as the adaptive method, the non-adaptive method is parallelized using the MW paradigm to realize load-balancing among workers [6]. As we mentioned in Section 3.2, the grain size of a task was experimentally selected to yield the highest performance: 50 and 10,000 configurations per task for the adaptive and non-adaptive methods, respectively.

Table 1 shows the execution time for fine-grained ROM estimation with $360 \times 180 \times 360$ stance configurations. As compared with the parallel non-adaptive method, the parallel adaptive method reduces the execution time on Myrinet from 141 to 19.6 seconds and that on Ethernet from 197 to 26.8 seconds. These timing results, less than a half minute,

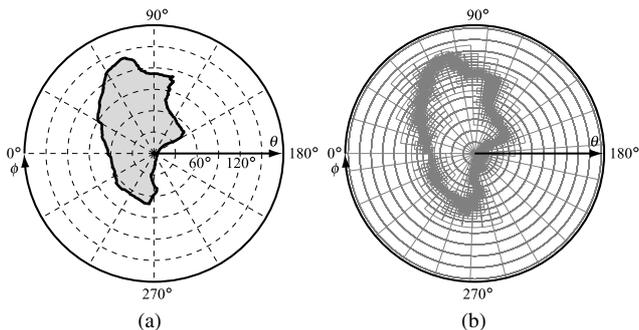


Fig. 8 (a) A slice of computed safe ROM, where $\psi = 0^\circ$, and (b) stance configurations sampled by the adaptive method, where sampled configurations are shown as lattice points of the mesh.

are acceptable for intraoperative surgical planning. Since the sequential adaptive method takes 1263 seconds, parallelization, as well as AMR, is a key strategy for intraoperative assistances.

Figure 8 shows stance configurations sampled by the adaptive and non-adaptive methods, presenting how the adaptive method reduces the amount of computation. We can see that the adaptive method investigates in detail the stance configurations only around the boundary of the ROM. This reduces the number of investigated configurations from 23,328,000 to 1,594,816 configurations, and thereby the speedup to the non-adaptive method reaches a factor of 12.9 ($= 16,324/1263$) on a single CPU machine.

In addition to the timing benefit, the adaptive method allows progressive visualization, as we mentioned in Section 3.1. For example, the method roughly outlined the safe ROM using sampling intervals of 16° , 8° , and 4° after 6.1, 6.1, and 6.6 seconds on Myrinet, respectively. These results mean that progressive visualization is not smoothly presented to users. This is due to the initialization of V-COLLIDE, which takes 6 seconds to construct data structure from the input polygons. Thus, although V-COLLIDE requires pre-processing time for rapid CDs, we think that progressive visualization is necessary for intraoperative estimation to report the progress of ROM estimation to surgeons.

To evaluate the scalability of our method, we measured the speedup on different numbers of CPUs (Fig. 9). Here, the speedup is the ratio of the sequential execution time to the parallel execution time. While the speedup of non-adaptive method linearly increases, that of adaptive method saturates as the number of CPUs increases. This is mainly due to the less parallelism in the adaptive method as compared with the non-adaptive method. That is, although the adaptive method reduces the amount of computation, it loses the complete independence of tasks that the non-adaptive method has. Furthermore, the adaptive method needs the master to compute the status for stance configurations. Thus, less parallelism and additional work decrease the speedup of the adaptive method.

Note here that although the non-adaptive method provides higher speedup than the adaptive method, it

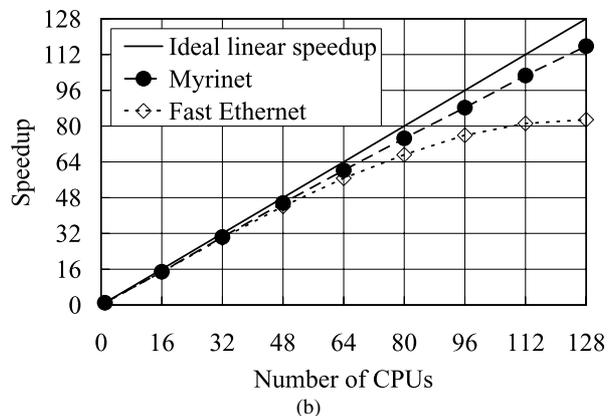
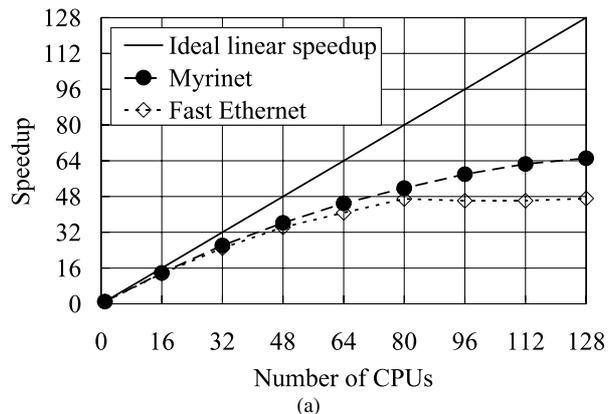


Fig. 9 Speedups of (a) adaptive method and (b) non-adaptive method.

takes approximately seven times longer execution time on 128 CPUs. Therefore, speedup is not always a good measure for the evaluation of parallel methods.

Next, we show the effect of the asynchronous mechanism presented in Sect. 3.2. Figure 10 shows the behaviors of the master and workers in timeline. There is apparently long waiting time in the synchronous implementation. As we mentioned in Section 3.2, this waiting time is due to the master, which samples configurations before increasing the estimation level. In contrast, the waiting time is eliminated by our asynchronous mechanism.

Actually, as compared with a synchronous implementation, our asynchronous implementation reduces the execution time from 21.3 to 19.4 seconds and from 53.9 to 26.8 seconds on Myrinet and on Ethernet, respectively. Thus, our mechanism demonstrates more reduction on high-latency, low-bandwidth network. This is due to the communication overhead incurred at the master. That is, relatively larger overhead makes it easier for the master to quickly assign tasks to workers, because the master has more time to compute the status of returned configurations and to sample new configurations around them. Thus, eliminating the waiting time is necessary to obtain higher efficiency, especially for high-latency, low-bandwidth networks.

Finally, we compare the safe ROMs estimated by our adaptive algorithm and the non-adaptive algorithm. As we mentioned in Section 3.1, two algorithms may produce dif-

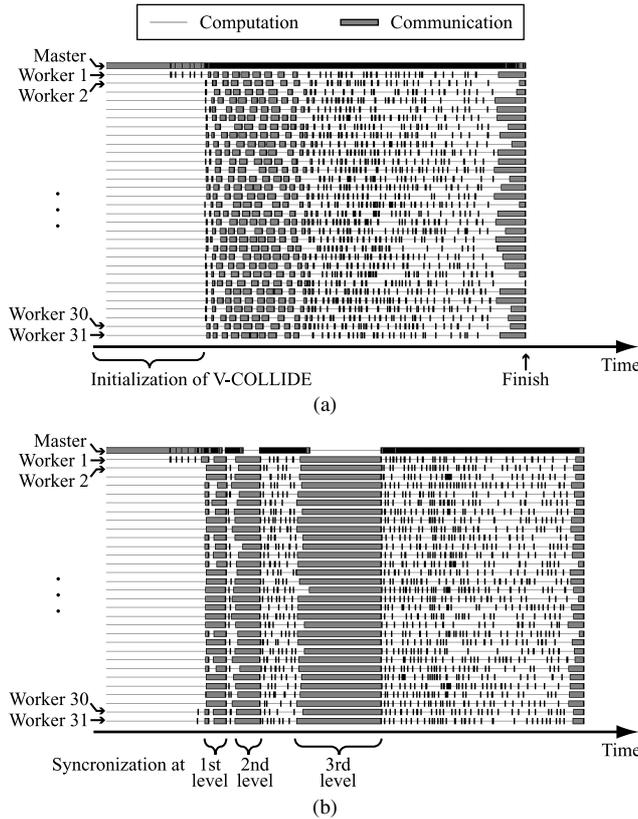


Fig. 10 Timeline views of (a) asynchronous implementation and (b) synchronous implementation, showing the process behavior until the 4th level on 16 PCs, each with 2 CPUs. Computation part includes initialization, task processing and status control. Communication part corresponds to data transfer and waiting time for incoming data. Worker 1 finishes V-COLLIDE initialization earlier than others, because it runs on the same PC as the master, which does not need to initialize V-COLLIDE. Other workers take relatively longer time for initialization, mainly due to I/O and bus contention.

ferent results if the initial sampling intervals are excessively large. For the datasets used in the experiments, we obtain the same ROMs if the initial sampling intervals are smaller than 16° . Otherwise, we have different ROMs.

In the latter cases, the adaptive algorithm fails to refine the meshes around configurations close to the boundary. This failure is due to the same reason mentioned in Section 3.1, that is, the same situation occurs at each estimation level. If the initial sampling intervals are larger than 32° , the algorithm decides not to refine the mesh around the raised portions of the boundary, such as the lower part of the ROM in Fig. 8 (a), that surrounded by collision-free configurations sampled by the coarse mesh. In contrast, the algorithm successfully refines the mesh around the raised portions if the initial intervals are smaller than 16° . Thus, the smaller intervals can avoid the failure of the adaptive algorithm.

6. Related Work

Earlier systems [17]–[19] compute the safe ROM in $O(1)$ time, but they do not take account of bone-implant and

bone-bone impingements. These systems assume that the hip movement is limited only by implant-implant impingements, such as cup-neck impingements. Under this assumption, the limitation of joint movement can be easily estimated by solving simple formulas, because cups and necks have simple shapes such as spheres and cylinders.

To tackle more practical cases, recent systems [3], [4] estimate more accurate ROMs by taking account of all of the impingements mentioned above. However, their detailed approach involves a large amount of computation due to the high cost of CDs between complex-shaped bones. This is a critical problem for intraoperative estimation, where the estimation must be rapidly done in a half minute. Therefore, such systems are required to reduce the computational amount by degrading the sampling intervals or the dimension of the ROM.

Kawasaki et al. [6] have developed a cluster system that parallelizes Sato’s system [3]. However, their cluster system limits the number of stance configurations because their non-adaptive method is not fast enough for intraoperative estimation. To the best of our knowledge, there is no work achieving complexity reduction for ROM estimation.

In the research area of robotics, there are many works [20]–[22] dealing with a problem similar to the estimation of hip joint ROM. Their problem is to find collision-free robot configurations, such as angular displacements of manipulator joints, which keep the robot free from collisions with stationary obstacles. Solving this problem is essential to plan a robot motion such that the robot could move from a start to a destination without collisions.

The robotics methods are applicable to our problem by assuming that a pelvis is an obstacle and a femur is a robot manipulator with three degrees of freedom (DOF). However, they are not optimized well for our medical problems where objects have complex shapes and obstacles are limited by a single object. In contrast, the robotics methods are optimized to deal with moving a simple-shaped robot through many simple-shaped obstacles. Even if robots and obstacles have complex shapes, such methods are allowed to substitute simple-shaped objects for robots and obstacles, because exact ROMs are not required for computing the robot motion path. In contrast, ROM estimation methods employ a prune-based CD algorithm in order to obtain exact ROMs in rapid time for such complex problems.

7. Conclusions

We have presented a ROM estimation method that is rapid enough for assisting the surgeon during THR surgery. Our method achieves fast estimation based on two strategies: the AMR strategy where the safe ROM is non-uniformly refined in a coarse-to-fine manner to reduce the computational complexity; and the parallelization strategy where stance configurations are investigated in simultaneously and asynchronously to obtain further acceleration.

We have shown that the AMR strategy reduces the time complexity under the assumption that the ROM is a single

closed region. Furthermore, the experimental results indicate that the implementation on a cluster of 64 PCs achieves estimation of $360 \times 360 \times 180$ stance configurations within a half minute, and thereby plays a key role in selecting and aligning the optimal combination of artificial joint components during surgery.

In future work, we will parallelize V-COLLIDE initialization in order to achieve higher speedup on many CPUs. We are also planning to verify the performance for various clinical datasets.

Acknowledgment

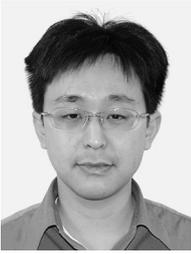
This work was partly supported by JSPS Grant-in-Aid for Scientific Research on Priority Areas (17032007) and for Scientific Research (B)(2)(18300009), and the Sasakawa Scientific Research Grant from The Japan Science Society. We are also grateful to the anonymous reviewers for their valuable comments.

References

- [1] A.M. DiGioia, B. Jaramaz, M. Blackwell, D.A. Simon, F. Morgan, J.E. Moody, C. Nikou, B.D. Colgan, C.A. Aston, R.S. Labarca, E. Kischell, and T. Kanade, "Image guided navigation system to measure intraoperatively acetabular implant alignment," *Clinical Orthopaedics and Related Research*, vol.355, pp.8–22, Oct. 1998.
- [2] B. Jaramaz, A.M. DiGioia, M. Blackwell, and C. Nikou, "Computer assisted measurement of cup placement in total hip replacement," *Clinical Orthopaedics and Related Research*, vol.354, pp.70–81, Sept. 1998.
- [3] Y. Sato, T. Sasama, N. Sugano, K. Nakahodo, T. Nishii, K. Ozono, K. Yonenobu, T. Ochi, and S. Tamura, "Intraoperative simulation and planning using a combined acetabular and femoral (CAF) navigation system for total hip replacement," *Proc. 3rd Int'l Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI'00)*, pp.1114–1125, Pittsburgh, PA, USA, Oct. 2000.
- [4] Y. Otake, N. Suzuki, A. Hattori, K. Hagio, N. Sugano, K. Yonenobu, and T. Ochi, "Four-dimensional model of the lower extremity after total hip arthroplasty," *J. Biomechanics*, vol.38, no.12, pp.2397–2405, Dec. 2005.
- [5] J.A. Richolt, M. Teschner, P.C. Everett, M.B. Millis, and R. Kikinis, "Impingement simulation of the hip in SCFE using 3D models," *Computer Aided Surgery*, vol.4, no.3, pp.144–151, Oct. 1999.
- [6] Y. Kawasaki, F. Ino, Y. Mizutani, N. Fujimoto, T. Sasama, Y. Sato, N. Sugano, S. Tamura, and K. Hagihara, "High-performance computing service over the Internet for intraoperative image processing," *IEEE Trans. Inf. Technol. Biomed.*, vol.8, no.1, pp.36–46, March 2004.
- [7] R. Buyya, ed., *High Performance Cluster Computing*, Prentice Hall PTR, Englewood Cliffs, NJ, 1999.
- [8] S. Gottschalk, M.C. Lin, and D. Manocha, "OBBTree: A hierarchical structure for rapid interference detection," *Computer Graphics (Proc. ACM SIGGRAPH'96)*, vol.30, no.3, pp.171–180, Aug. 1996.
- [9] P.M. Hubbard, "Collision detection for interactive graphics applications," *IEEE Trans. Vis. Comput. Graphics*, vol.1, no.3, pp.218–230, Sept. 1995.
- [10] J.T. Klosowski, M. Held, J.S. Mitchell, H. Sowizral, and K. Zikan, "Efficient collision detection using bounding volume hierarchies of k -DOPs," *IEEE Trans. Vis. Comput. Graphics*, vol.4, no.1, pp.21–36, Jan./March 1998.
- [11] T.C. Hudson, M.C. Lin, J. Cohen, S. Gottschalk, and D. Manocha, "V-COLLIDE: Accelerated collision detection for VRML," *Proc. 2nd Symp. Virtual Reality Modeling Language (VRML'97)*, pp.117–124, Monterey, CA, USA, Feb. 1997.
- [12] M.D. Berg, M. Overmars, O. Schwarzkopf, and M.V. Kreveld, "Robot motion planning: Getting where you want to be," in *Computational Geometry: Algorithms and Applications*, 2nd ed., pp.267–290, Springer-Verlag, Berlin, Germany, 2000.
- [13] K.J. Falconer, *Fractal Geometry: Mathematical Foundations and Applications*, John Wiley and Sons, Chichester, NY, USA, 1990.
- [14] N.J. Boden, D. Cohen, R.E. Felderman, A.E. Kulawik, C.L. Seitz, J.N. Seizovic, and W.K. Su, "Myrinet: A gigabit-per-second local-area network," *IEEE Micro*, vol.15, no.1, pp.29–36, Feb. 1995.
- [15] F. O'Carroll, H. Tezuka, A. Hori, and Y. Ishikawa, "The design and implementation of zero copy MPI using commodity hardware with a high performance network," *Proc. 12th ACM Int'l Conf. Supercomputing (ICS'98)*, pp.243–250, Melbourne, Australia, July 1998.
- [16] Message Passing Interface Forum, "MPI: A message-passing interface standard," *Int'l J. Supercomputer Applications and High Performance Computing*, vol.8, no.3/4, pp.159–416, 1994.
- [17] B. Jaramaz, C. Nikou, D.A. Simon, and A.M. DiGioia, "Range of motion after total hip arthroplasty: Experimental verification of the analytical simulator," *Proc. 1st Joint Conf. Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVRMed-MRCAS'97)*, pp.573–582, Grenoble, France, March 1997.
- [18] M. Seki, N. Yuasa, and K. Ohkuni, "Analysis of optimal range of socket orientations in total hip arthroplasty with use of computer-aided design simulation," *J. Orthopaedic Research*, vol.16, no.4, pp.513–517, July 1998.
- [19] D.D. D'lima, A.G. Urquhart, K.O. Buehler, R.H. Walker, and C.W. Colwell, "The effect of the orientation of the acetabular and femoral components on the range of motion of the hip at different head-neck ratios," *J. Bone and Joint Surgery*, vol.82, no.3, pp.315–321, March 2000.
- [20] T. Lozano-Pérez, "A simple motion planning algorithm for general robot manipulators," *IEEE J. Robotics and Automation*, vol. RA-3, no.3, pp.224–238, June 1987.
- [21] W.S. Newman and M.S. Branicky, "Real-time configuration space transforms for obstacle avoidance," *Int. J. Robot. Res.*, vol.10, no.6, pp.650–667, Dec. 1991.
- [22] B. Curto, V. Moreno, and F.J. Blanco, "A general method for C-space evaluation and its application to articulated robots," *IEEE Trans. Robot. Autom.*, vol.18, no.1, pp.24–31, Feb. 2002.



Yasuhiro Kawasaki received the B.E. and M.E. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 2002 and 2004, respectively. He is currently working toward the Ph.D. degree at the Department of Computer Science, Graduate School of Information Science and Technology, Osaka University. His current research interests include high performance computing, Grid computing, and systems architecture and design. He received the Best Paper Award at the 10th International Conference on High Performance Computing (HiPC'03).



Fumihiko Ino received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1998, 2000, and 2004, respectively. He is currently an Assistant Professor in the Graduate School of Information Science and Technology at Osaka University. His research interests include parallel and distributed systems, software development tools, and performance evaluation. He received the Best Paper Award at the 2003 International Conference on High Performance

Computing (HiPC'03) and the Best Paper Award at the 2004 Symposium on Advanced Computing Systems and Infrastructures (SACIS'04).



Yoshinobu Sato received the B.S., M.S., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1982, 1984, 1988 respectively. From 1988 to 1992, he was a Research Engineer at the NTT Human Interface Laboratories, Yokosuka, Japan. In 1992, he joined the Division of Functional Diagnostic Imaging of Osaka University Medical School as a Faculty Member. From 1996 to 1997, he was a Research Fellow in the Surgical Planning Laboratory, Harvard Medical

School and Brigham and Women's Hospital, Boston, MA. He is currently an Associate Professor in the Graduate School of Medicine and Graduate School of Engineering Science at Osaka University, where he leads a group conducting research on 3-D image analysis and surgical navigation systems in the Division of Interdisciplinary Image Analysis.



Shinichi Tamura received the B.S., M.S., and Ph.D. degrees in electrical engineering from Osaka University, Osaka, Japan, in 1966, 1968 and 1971, respectively. He is currently a Professor of Graduate School of Medicine and Graduate School of Engineering Science of Osaka University. He has published over 200 papers in scientific journals and received several paper awards from journals including Pattern Recognition and Investigative Radiology. His current research activities include works in the field of

medical image processing and its applications. Dr. Tamura is currently an Associate Editor of Pattern Recognition and a Vice Editor-in-Chief of Medical Imaging Technology. He is a member of the Information Processing Society of Japan, the Japanese Society of Medical Imaging Technology, and the Japan Radiological Society.



Kenichi Hagihara received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Osaka, Japan, in 1974, 1976, and 1979, respectively. From 1994 to 2002, he was a Professor in the Department of Informatics and Mathematical Science, Graduate School of Engineering Science, Osaka University. Since 2002, he has been a Professor in the Department of Computer Science, Graduate School of Information Science and Technology, Osaka University. From 1992 to 1993, he

was a Visiting Researcher at the University of Maryland. His research interests include the fundamentals and practical application of parallel processing. He received the Best Paper Award at the 2003 International Conference on High Performance Computing (HiPC'03) and the Best Paper Award at the 2004 Symposium on Advanced Computing Systems and Infrastructures (SACIS'04).