07(XV)-78 Topology Independent Model for Medical Robotic Simulation

C. Chui[a], I. Sakuma[a]
[a] *Department of Mechanical Engineering, National University of Singapore, Singapore*
[b] *Department of Precision Engineering, The University of Tokyo, Tokyo, Japan*

Abstract: The field of medical robotic is relatively new, and computer aided surgical system with robot for treatment delivery has not been fully exploited. An approach in advancing this field is via a more efficient preparation of the robotic surgical intervention using computer simulation. In the virtual medical robotic system that we are developing for fluoroscopic image guided liver-needle insertion and ablation, C-Arm as well as one or more needle insertion robots are represented as independent robotic models in one work cell. Unlike typical industrial robot, medical robot is operating on an unstructured clinical environment with many variations. In order to provide the computational power required for medical robotic simulation, we are exploring parallel computing using a microcomputer system with multiple core processors. Instead of relying on the task scheduler based upon load balancing, we investigated LogP which is a theoretical model that does not consider the topology of the communication network. The network is represented using parameters that account for limited bandwidth into and out of the processor, latency and overhead. We demonstrated that parallel algorithms can be derived and evaluated using this model.
Key words: Medical robot, Surgical simulation, Parallel processing, Theory of Computing

1. Introduction

By extending human surgeon's ability to plan and carry out surgical interventions more accurately and less invasively, computer aided surgical system including medical robotic, can greatly reduce costs, improve clinical outcomes, and improve the efficiency of health care delivery. The field of medical robotic is relatively new, and computer aided surgical system with robot for treatment delivery is far from being full exploited. A solution to remedy this discrepancy is through a more efficient preparation of the intervention via computer simulation. However, computer simulation of medical robotic is still very much in its infancy.

Medical robotic development systems including graphical simulation are of considerable interest [1], for several reasons. Such systems permit planning and evaluation of the robot and/or related surgical equipment before they are used in the surgery. They permit the programming of robot away from the clinical environment in which it must operate. The risk of damage is reduced if it is possible to visually inspect the execution of the task on a graphic screen before it is executed clinically. Nevertheless, this is a complex and computationally intensive process, and unlike industrial robotic system, we are working on an unstructured clinical environment with many variations that will demand frequent parameter changes and interactive simulation.

In our virtual medical robotic systems for fluoroscopic image guided needle insertion, the robot and C-arm are represented as independent robotic models in one work cell. An objective of the simulation is to derive with our conceptual robot the optimal needle placement and insertion with unobstructed view from C-Arm and minimal X-ray dosage.

Despite the recent advancement of parallel computing hardware such as workstation cluster, and recently, microcomputer system with multiple core processors, it remains an issue on how to fully utilize this computing hardware to deliver optimal performance. In this paper, we describe our investigation on two theoretical models for cluster and multicore computing respectively for medical robotic simulation.

2. The LogP Model

In LogP model [2, 3], communication between processors is represented by a set of parameters, and through a broadcasting media. The LogP model is originally targeted as a model for multiprocessor systems, and is independent of the communication network topology. We want to use the LogP model to design and evaluate parallel/distributed algorithms for real time simulation of robot operations.

The LogP model we use contains the following components:

- P processors that run asynchronously and independently.
- An interconnection network whose topology is of no concern to us, except that we know:
  - g, a single parameter that accounts for limited bandwidth into and out of the processor. It is the rate at which a processor can send words into the network when it is ready to do so.
  - L is the latency.
  - o is the overhead, accounting for all other time in the execution of the remote instruction during which the processor is unavailable for computing.

g approximates $(1/B)$ where B is the bandwidth of the entire network. If an instruction is executed at a source at time t, the solution is not available at the destination until

$t + L$. The model is useful when it can be used to design and devaluate a large class of algorithms related to robotic simulation.

3. Performance Evaluation using LogP Model

We will start by proving the desired performance of this model using one-to-all broadcasting in Lemma 1, and all-to-all broadcasting in Lemma 2. One-to-all and all-to-all broadcasting are fundamental operations that appear many time as subtasks in medical robotic simulation. Note that g is the most important factor considered in our theoretical analysis, and to transmit n elements from one processor to another processor requires O(ng) time.

Lemma 1: One-to-All Broadcast

If one processor needs to send n elements to all other P-1 processors, it can be done in O($ng\log P$) time.

Suppose that there are P processors. Any one processor p wants to send a vector **b** of n elements to all other processors as in procedure One-to-All Broadcast. In this procedure, we may conceptually regard the execution as a binary tree, and the execution starts in top-down mode.

For example, if processor P_0 is to broadcast n elements to all other processors, it will send these elements to processors P_1 and P_2. They will then transmit these elements to P_3, P_4 and P_5, P_6 respectively in a concurrent manner. This process repeats until all processors receive the elements. The number of recursion equals to the depth of the tree. The height of the tree is O(log P). It is clear that the time complexity is O(ng log P).

Note that we can improve the O(ng log P) time complexity to O(ng + $g$log P) by pipelining the transmissions of the n elements.

Lemma 2: All-to-All Broadcast

If all processors need to send n elements to all other processors, and there are P processors, it can be done in O(ng P) time with no conflict.

Suppose that there are P processors, and each processor wants to send a vector b to all other processors.It is obvious that this will require O(ng P log P) time if we will to use procedure One-to-All Broadcast P times to accomplish the task. A more natural way is given in procedure All-to-All Broadcast. It is clear that there is no conflict in communication and O(ng P) time is required for this procedure.

Lemma 1 and Lemma 2 provide the theoretical basis to derive and evaluate theoretical optimal algorithms for other parallel scalar and matrix operations

```
Procedure One-to-All Broadcast {
   If all processors receive b then stop
   Else recursively transmit to the next two
   processors that do not yet receive b
}
Procedure All-to-All Broadcast{
   For k=1 to P-1
      For each processor p do in parallel
         Send b to processor (p+k) mod P
         Receive b from processor (p-k) mod
         P
}
```

Fig.1 Broadcast algorithms

4. Discussion and Conclusion

It is difficult is achieve optimal performance on parallel/distributed computing system using typical tasks scheduling methods based on load balancing. In this paper, we described our investigation on the application of LogP model which is a topology independent model, in derivation and evaluation of parallel algorithms for medical robotic simulation. Our work on the LogP model focuses on efficient computation. This is different from previous works that concentrated primarily on communication.

Reference

1) Falk V, F Mourgues et al (2005). Cardio navigation: planning, simulation, and augmented reality in robotic assisted endoscopic bypass grafting, Ann Thorac Surg 79(6): 2040-2047
2) Culler D, R Karp et al (1993). LogP: towards a realistic model of parallel computation, Principles Practice of Parallel Programming, pp. 1-12.
3) Chui C, S Jain et al (1994). Research activities in accelerated computing and theory program (ACT program), J High Performance Computing 1(1): 50-57.

原著

狭隘術野での精密低侵襲手術を支援するマニピュレータシステムの開発

Development of Manipulator System to Support Precise and Minimally-Invasive Surgery in a Small Operative field

西澤幸司(正会員)[ab]*, 村垣善浩[c], 藤江正克[d], 佐久間一郎[e], 伊関洋[c]
[a]東京女子医科大学大学院　先端生命医科学系専攻
[b]株式会社日立製作所　機械研究所
[c]東京女子医科大学　先端生命医科学研究所
[d]早稲田大学　理工学術院
[e]東京大学大学院　工学系研究科

Koji Nishizawa[ab], Yoshihiro Muragaki[c], Masakatsu G.Fujie[d], Ichiro Sakuma[e], Hiroshi Iseki[c]
[a] Graduate School of Medicine, Tokyo Women's Medical University.
[b] Mechanical Engineering Research Laboratory, Hitachi, Ltd.
[c] Institute of Advanced Biomedical Engineering & Science, Tokyo Women's Medical University.
[d] Faculty of Science and Engineering, Waseda University.
[e] Graduate School of Engineering, The University of Tokyo.

Abstract

A surgical manipulator system that can answer a wide range of surgical needs is described. We have ever developed a surgical manipulator system HUMAN (Hyper-Utility Mechatronic AssistaNt). It provides minimally invasive neurosurgery in a clinical setting. Besides the obvious restriction of only being able to use this system on the brain, the operative field that this system could treat was circular within a diameter of 1 cm. We therefore developed an advanced system based on HUMAN, which we call AMATERAS. This new surgical manipulator system enables precise and minimally invasive surgery on not only the brain but also on such areas as the abdomen. The setting of the manipulator needs to be usable for more than the brain and to have a treatable operative field of 3 cm minimum. We developed two new mechanisms: a positioning arm that has a five-bar linkage mechanism for pivoting the manipulators and a mechanism that puts some manipulators on a stand for an operating microscope, one that does not hit the surgical bed or the patient's body. With these functions, AMATERAS provides precise and minimally invasive surgery for many area of the body.

Key words

Surgical Support System, Clinical Application, Surgical Manipulator System.

*東京女子医科大学 先端生命医科学研究所
〒162-8666 東京都新宿区河田町 8-1
knishizawa@abmes.twmu.ac.jp
受付 2006 年 11 月 8 日; 採択 2007 年 5 月 7 日

1. 緒論

低侵襲手術のひとつである内視鏡下手術は，1985年にMuheが初めて腹腔鏡下胆嚢摘出術に成功[1]して以来，急速に広まった．内視鏡下手術は，低侵襲というメリットに加えて，従来の開腹術や小開腹術と比較しても，合併症は少ないことがSheaら[2]により示された．現在では，進行癌などの症例への展開や，心臓外科や婦人科などの消化器外科以外の診療科でも実施されるようになった．

しかし，内視鏡下で行なう治療には，低侵襲という利点の一方で，術具操作に熟練を要するという課題がある．そのため，治療を担当する執刀医の技量に応じて，患者が受けられる治療レベルに格差が生じるという一面も存在する．

近年では，術具の操作環境を改善し，より安全に低侵襲手術を実施する手段のひとつとして，手術用マニピュレータが注目されている．その一例として，欧米を中心に臨床応用されているda Vinci®やZEUS®があり，高い操作性が臨床現場において一定の評価を受けている[3)-5)]．国内では，腹腔手術用術具の操作性向上を目指して，神野らが開発したロボット鉗子[6]や，低侵襲に加えて微細な治療の実現を目的に，筆者らが開発した脳神経外科手術支援用マニピュレータシステムHUMAN[7]（以降，HUMANシステムと呼ぶ）等がある．HUMANシステムは，脳神経外科の分野では世界に先駆けて，2002年8月から，信州大学で臨床適用を実施した[8)9)]．その後に実施した臨床では，直径30mm程の小さな開頭からマニピュレータを挿入させて実施した，第三脳室底開窓術などを成功させている．

このような背景のもと，本研究は，脳神経外科に限らず，患者の全身を対象とするより多くの診療科における，内視鏡下での術具操作性を向上させ，安全で正確な治療を支援することを目的としている．その手段として，HUMANシステムの機構改良を行い，汎用的な高機能術具として，システムの適用領域を拡大する．このHUMANシステムの改良型を，手術支援マニピュレータシステム“AMATERAS”（以降，AMATERASシステム）[10]と呼ぶ．本論文では，HUMANシステムをほかの診療科に適用するための技術課題を示し，この結果に基づいて開発したAMATERASシステムの機能について検証する．

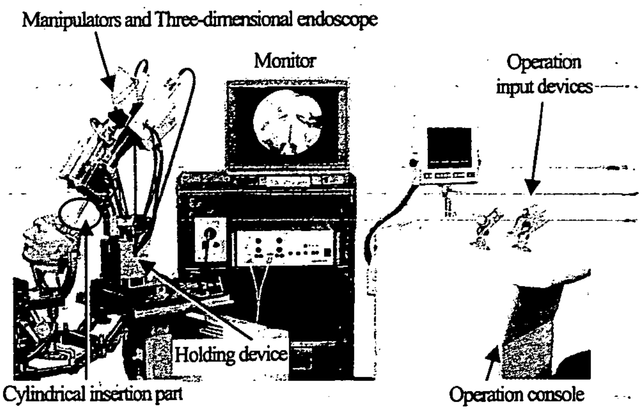

Fig. 1 HUMAN manipulator system.

2. AMATERASシステムの開発コンセプト

2. 1 HUMANシステムの機能

AMATERASシステムの基本機構となるHUMANシステム[7]（**Fig. 1**）は，内視鏡の映像をモニタで確認しながら，操作入力装置を用いて，保持装置に保持されたマニピュレータを遠隔操作する装置である．このシステムは，10μm以下の動作分解能を持つ3本のマニピュレータと内視鏡を，互いに干渉することなく直径10mmの挿入部（Fig. 1左，および**Fig. 2**）に束ねる機構を開発したことで，第三脳室底開窓術で実施したような小さな開頭部からの緻密な治療作業を実現した．

マニピュレータは，Fig. 2のα（首振り），β（回転），Z（並進）の三自由度動作が可能である．術具はFig. 2のa（開閉）とb（関節に対する回転）の二自由度動作が可能である．

マニピュレータと術具とは**Fig. 3**に示すユニットとして開発した．Fig. 2の挿入部に束ねられた3本のマニピュレータは，1本ごとにシステムへの着脱が自在である．術具もマニピュレータに着脱自在なユニットである．マニピュレータの後端から中空部を通して挿入し，マニピュレータと一体化させる機構を実現している．

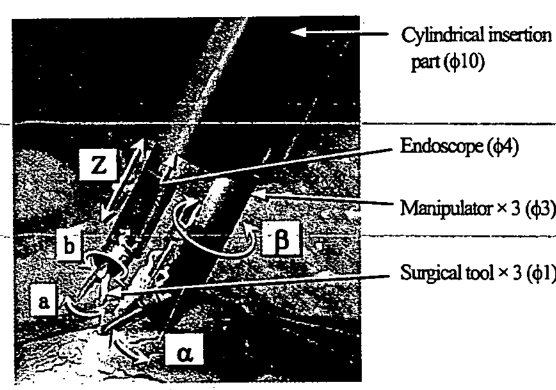

Fig. 2 Magnified image of cylindrical insertion part.

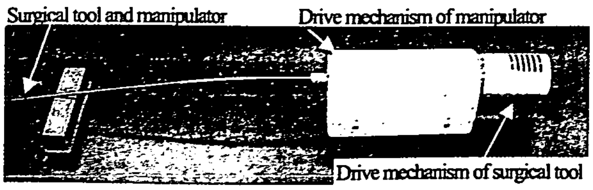

Fig. 3 Manipulator unit and surgical tool unit of HUMAN system.

2. 2 AMATERASシステムの適用対象

AMATARAS システムの開発では，HUMAN システムを，脳神経外科以外にも展開し，より多くの症例に適用させるための機能拡張を行なう.

AMATERAS システムが臨床現場に提供する付加価値は，“精密と低侵襲”である. 具体的には，患者の全身を対象に，人の手や従来の術具では実施できないような，緻密で正確な治療作業や，少数の小切開部からの治療の実現である.

そのため，適用対象と考えているのは，細い術具や挿入経路が制限される症例，狭い範囲での精密な治療を必要とする症例などである. 例えば，眼底や口腔や鼻腔など，術具のアクセス経路や術野が制限される治療を始め，子宮へのダメージを最小限に抑えながら行なう，胎児治療等の症例や，腱や神経の縫合などの微細操作には有効である.

HUMAN システムは脳神経外科のみを対象としていたため，手術機能を限定している. AMATERAS システムが対象とする，より多くの症例では，症例に応じてマニピュレータや術具に求められる仕様（マニピュレータの可動範囲やサイズ，駆動部出力，動作速度，術具の形状や機能など）が異なる. そのため，本システムでは，症例，術式に応じて最適な仕様に修正したユニットを開発し，手術ごとに最適なユニットを装備して治療を行なうことが必要である. 本システムの基本機構は，2.1 項で述べたように，マニピュレータと術具をユニット構造として開発しているため，着脱部の互換性をとることで，症例に合わせたユニットをシステムへ装備できる.

3. 課題

HUMAN システムのマニピュレータユニットは仕様変更の自由度は高いが，システム全体としては，(a)患部に対するセッティング機能と，(b)治療できる領域（術野）の広さ，という二つの機能拡張が必要である.

(a) セッティング機能

従来のHUMANシステムの保持装置は，マニピュレ

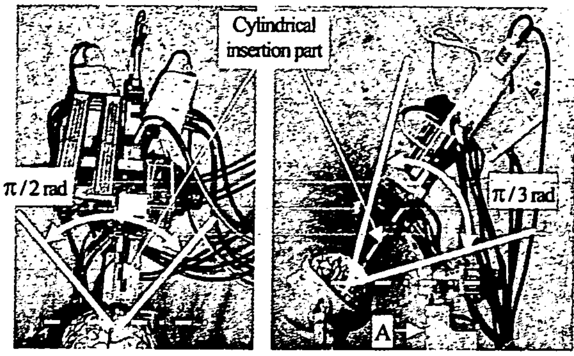

Fig. 4 Range of holding device movement.

ータをセッティングできる領域が限定されている. 当初，この保持装置に求められた可動範囲は，Fig. 4に示すヨー角$\pi/2$ rad，ピッチ角$\pi/3$ radの角度で囲まれる領域であった. 患者に応じて，保持装置の設置場所自体の調整が可能であり，かつ，保持装置が大型化すると，運用上のデメリットが発生することから，可動範囲をFig. 4の領域に限定して保持装置の小型化を図った. さらに，安全確保のために，頭部に挿入している挿入部の姿勢は，マニピュレータを操作している外科医からは操作できない仕様とした. しかし，脳神経外科以外での臨床適用を前提に考えると，Fig. 4のAで示す部位が，患者やベッドと干渉するなど，アプローチできる範囲は制限を受けることとなる. 例えば，鼻腔へのアプローチでは，Fig. 5に示すように，患者の胸部側から鼻腔へ向けてマニピュレータをセッティングする必要がある. しかし，従来の保持装置では，このように，患者の内側から外側に向かうような姿勢でのセッティングは不可能である. さらに，胎児治療などを行なう場合でも，子宮へマニピュレータをアプローチさせる際に，患者の身体と保持装置とが干渉する.

(b)治療できる領域の広さ

治療できる領域（術野）を治療可能領域と呼び，3本のマニピュレータのうち少なくとも 2 本がアプローチできる領域と定義する. その領域の広さは，実用上の観点から，領域内に内接する円領域の直径で表わす.

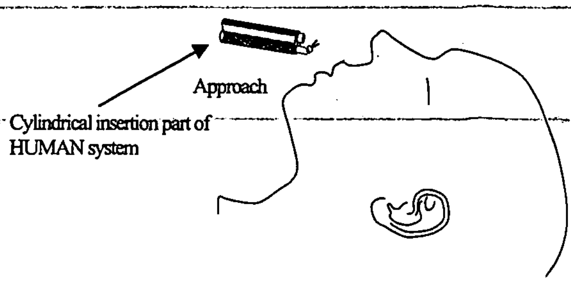

Fig. 5 Approach posture to a nasal cavity.

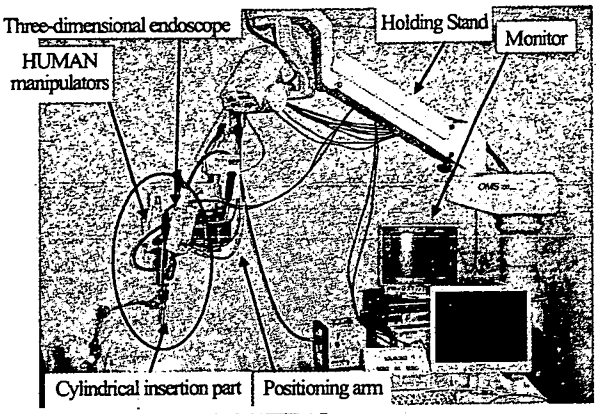

Fig. 6 AMATERAS system prototype.

従来の HUMAN システムが対象とする腫瘍は，直径数ミリの大きさであるため，治療可能領域を，直径 10mm に限定していた．これは，微小な開頭からの，人の手を越える微細治療を目的とする HUMAN システムにとっては，充分な広さであった．

AMATERASシステムでは，少なくとも直径 30mmの治療可能領域が必要であると考えている．この広さは，本システムの対象が，2 章で述べたように，細い術具や挿入経路が制限される症例，狭い範囲での精密な治療を必要とする症例などであることから，妥当である．その一例として，胎児の二分脊椎症などは，患部の大きさが 10mm四方程度であり，30mm角の術野を確保するスタビライザの開発等も試みられている[11]．

4. AMATERAS システムの機構

今回，新たに開発したAMATERASシステムの外観を**Fig. 6**に示す．セッティング機能の拡張と，治療可能領域の拡大という二つの技術課題に対して，手術顕微鏡用の保持スタンドにマニピュレータを搭載する保持機能と，マニピュレータの挿入姿勢を遠隔操作するポジショニングアーム機構とを新規に開発した．

本システムは，八自由度を持つ保持スタンドの先端に，ポジショニングアームを装備し，**Fig. 7**のように，ポジショニングアームの二自由度フリージョイントを介して，マニピュレータを装備する．さらに，マニピュレータの挿入部は，Fig. 7左下のトロカールに通し，トロカールを自在継手の三自由度ピボットジョイントで保持する構成である．

保持スタンドは，スタンド自身が手術ベッドの周囲を移動できる．さらに，ベッドと干渉することなく，天井方向から患者にマニピュレータをセッティングすることができる．セッティング時には，ハンドル（**Fig. 8**左）を握り，ロック解除ボタンを押すことで，Fig. 8中の 2～7で示す六自由度をフリーにできる．また，フットスイッチを用いてFig. 8中に 1,8 で示す自由度を電動操作し，マニピュレータの位置を微調整することもできる．これらの機能により，従来のHUMANシステムでは不可能であった身体部位への俊敏なセッティングを可能にした．

ポジショニングアームは，マニピュレータを保持・制御するリンク機構（Fig. 7）を有する[10]．このリンク機構を制御することで，フリージョイント部を，Fig. 7に示すXYの二方向に制御する．これにより，ピボットジョイント部分を回転中心とするピボット動作をさせて，マニピュレータの挿入姿勢を操作し，さらに，その姿勢のまま，マニピュレータ先端の三自由度動作による治療操作を実施する二重駆動システムを開発した．

マニピュレータの治療可能領域の中心点（点O）がポジショニングアームによって移動可能となる領域は，**Fig. 9**のような，半径 49mmと半径 25mmの円弧で囲まれる形状となる．点Oの初期位置は，白い同心円の中心点に一致する．このときの，二重駆動のひとつ目であるマニピュレータの治療可能領域は，直径 10mm

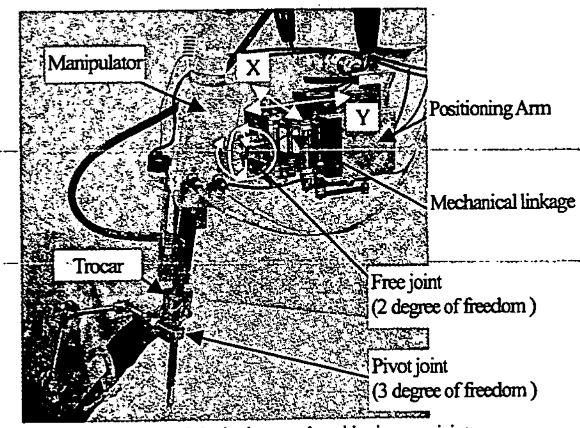

Fig. 7 Manipulator and positioning arm joint.

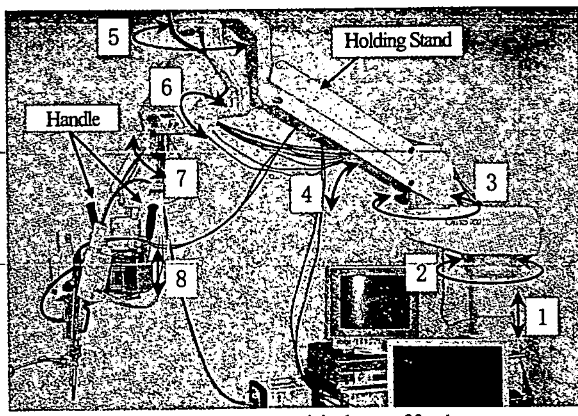

Fig. 8 Setting movement eight degree of freedom.

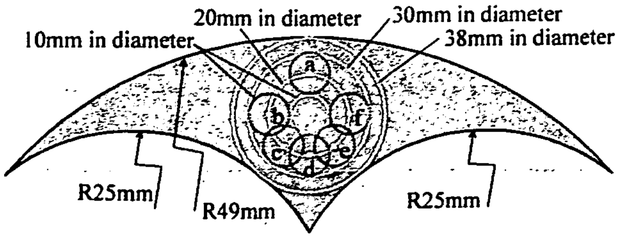

Fig. 9 Manipulator tip movement range.

の円である．直径 30mm の治療可能領域を実現するためには，二つ目の駆動機構であるポジショニングアームの動作により，点 O を図中の直径 20mm の範囲で操作できれば良い．点 O が境界上に位置するとき，その位置で，直径 10mm(例えば図中の a～fの黒い円)の範囲で治療できるため，二重駆動全体で直径 30mm の治療可能領域を得られる．本システムでは機構的余裕をとるため，直径 20mm の動作範囲に対して，直径 38mm まで移動可撓な設計とした．

5. 検証実験

5.1 治療可能領域に対する評価実験

1) 目的

本実験では，試作したポジショニングアームを制御し，直径 30mm 以上の治療可能領域が確保されていることを検証する．

2) 方法

直径 30mm以上の治療可能領域を確保するためには，挿入部の中心が，直径 20mm以上の円領域内を動作できればよい．そこで，初期位置から各方向へ 10mm以上移動できることを確認するために，**Fig. 10** に示すテストチャートを用いた計測を行った．テストチャートは，直径 1mm間隔で同心円を描いたものであり，チャート内には同心円の直径を記している．

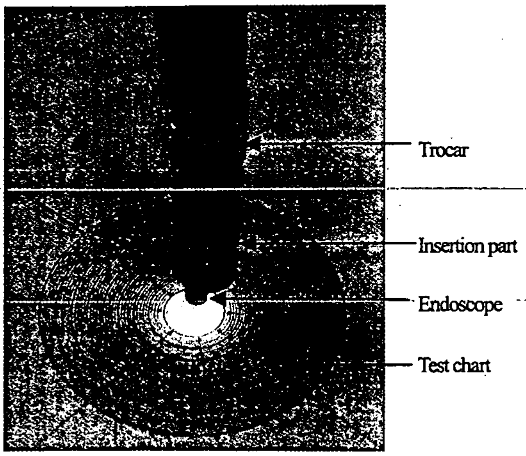

Fig. 10 Measurement experiment

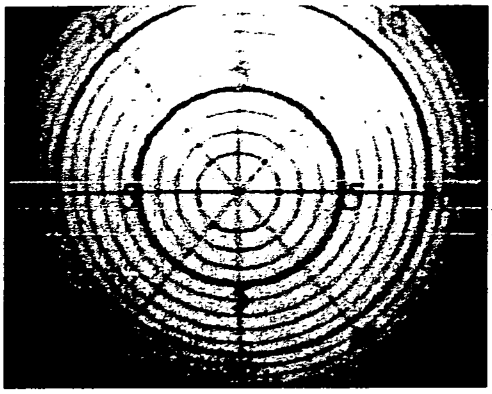

Fig. 11 Endoscopic image of test chart.

まず，内視鏡を装備した挿入部を，テストチャートへ垂直に，かつ，内視鏡画像(**Fig. 11**)の中心にテストチャートの中心が一致するようにセッティングする．

次に，内視鏡画像の中心が直径 20mm以上の円弧に沿うように，ポジショニングアームを動作させる．このとき，6 つの確認点(Fig. 9のa～f)が内視鏡画像の中心に移動した時点の映像を記録する．この確認点は，円弧状のπ/2 radごとの 4 点(a,b,d,f)，および，直径 20mmの円とグレーで示す移動可能領域の境界部分とが最も近づく点(a,c,e)を選んだ．

本実験では，内視鏡の視野の移動範囲を計測するが，内視鏡とマニピュレータが挿入部に装備される相対位置は固定されているため，本実験の計測結果は，二重駆動の一つ目であるマニピュレータの治療可能領域を，二つ目の駆動機構であるポジショニングアームが移動させた範囲と考えてよい．

ゆえに，本実験では，内視鏡画像の中心を対象として，視野の動作範囲を検証し，その結果をもって，治療可能領域の広さを検証する．

3) 結果

内視鏡画像により，挿入部が直径 20mm以上の円弧に沿って動作できることを確認した．その際の確認点a～fが内視鏡画像の中心に位置したときの様子を **Fig. 12**の(a)～(f)に示し，各画像の中心(＋印)からテストチャートの中心点までの距離を**Table 1**に纏める．

全ての点において，初期位置から 11mm の距離にある直径 22mm の円弧を画像中心に捉えることができた．これは，従来の 9 倍の広さとなる直径 30mm 以上の治療可能領域が実現できたことを示している．

Table 1 Distance from center of test chart.

	(a)	(b)	(c)	(d)	(e)	(f)
Distance [mm]	10.8	11.1	10.7	11.2	11.1	11.1

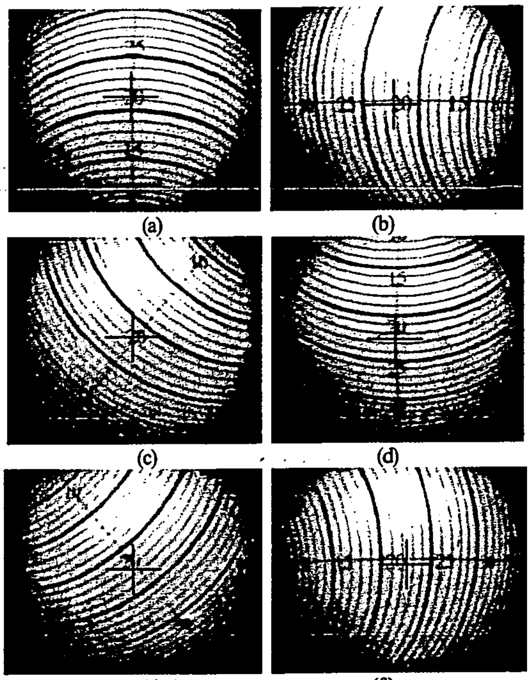

Fig. 12 Endoscopic image of test chart on positioning arm movement.

5．2 内視鏡的治療への適用性評価実験

1） 目的

本実験は，従来の HUMAN システムではセッティング困難であった部位へ AMATERAS システムをセッティングできること，および，その状態のまま目標値以上の治療可能領域が確保できていることを確認する．

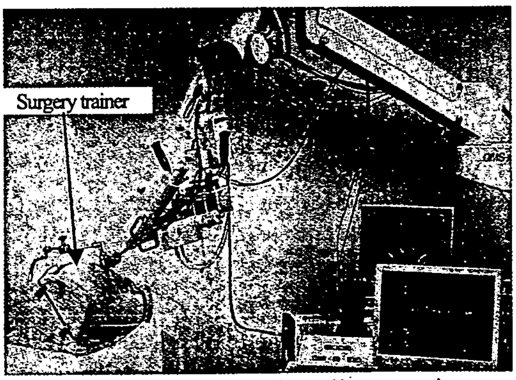

Fig. 13 View of manipulators inserted in surgery trainer.

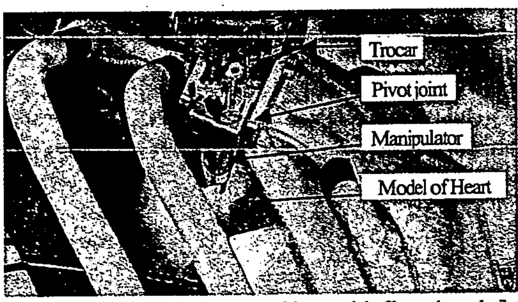

Fig. 14 View of manipulators approaching model of heart through ribs.

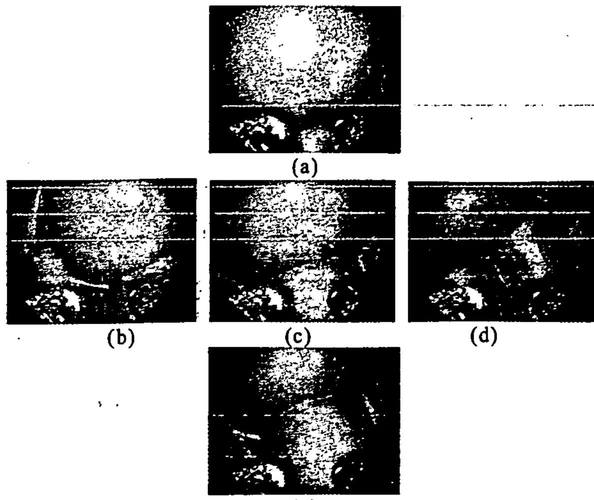

Fig. 15 Endoscopic image accompanying an action. (c) is initial position. (a) is upper, (b) is left, (d) is right and (e) is lower operation from (c).

2） 方法

低侵襲冠状動脈バイパス手術トレーナに対して，臨床を模擬してマニピュレータをセッティングし，ポジショニングアームを操作する．ピボット動作の支点となるピボットジョイントは体表位置にセッティングする．

本システムの治療可能領域は直径 30mm の円領域を最小限の目標値としているため，初期位置より 10mm 以上の距離を動作できればよい．実験中は動作距離を限定せずに操作し，実験後に内視鏡画像から動作距離を確認する．

3） 結果

手術トレーナの肋間から心臓に向けてセッティングしたマニピュレータの様子を**Fig. 13**，**Fig. 14**に示す．また，実験開始時の内視鏡画像を**Fig. 15** (c)に示し，ポジショニングアームを操作して，内視鏡を上下左右へ移動させたときの映像を同図(a),(b),(d),(e)に示す．各画像は，心臓のモデル上の**Fig. 16**に示す位置に対応している．図中の＋印は，移動後の各視野の中心位置である．(c)図の視野の中心位置から，移動後の各視野の中心位置までの距離を**Table 2**に纏める．

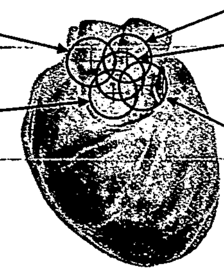

Fig. 16 Correspondence of endoscopic image and model of heart.

Table 2 Distance from center of endoscopic view.

	(c)→(a)	(c)→(b)	(c)→(d)	(c)→(e)
Distance [mm]	10.9	14.7	19.4	11.8

表中の(a)～(e)はFig. 15の(a)～(e)に対応している.

Fig. 13, Fig. 14に示すように, トロカールを通して, マニピュレータを胸腔内へ挿入するセッティングが可能であることを確認した. また, Fig. 15に示すように, 最初にセッティングした視野では見ることができない領域をポジショニングアーム操作で捕らえることが可能となったことも確認した.

肋間を通して患部にアプローチした状態での動作範囲は, Table 2に示すように, 目標値の10mmを超えた操作ができていることを確認した. 今回の実験で操作した治療可能領域は, 各視野の中心点(＋印)を通る楕円を想定し, その面積を計算すると, 従来の治療可能領域の15倍の広さとなる.

6. 考察

従来の腹腔手術に適用するマニピュレータは, ひとつの切開からひとつのマニピュレータ, あるいは, 内視鏡を挿入して治療を行なう. そのため, 切開数は4～5と多い. また, 大きく, 重い臓器を扱うため, 広い動作範囲と大きな出力を備えているのが特徴である.

本システムは, ひとつの切開から3本のマニピュレータと内視鏡を挿入して治療できるので, 従来より切開数を減らすことができ, 低侵襲性に優れている. また, 本システムの特徴を発揮できる対象は, 従来のマニピュレータと異なり, 狭い術野での精密・低侵襲な治療を必要とする症例である.

動作実験では, トロカールを通して患部へアプローチできたことで, 腹腔鏡下手術で一般的な気腹法に対応できることを示した. また, 肋間という制限されたアクセスルートを通した状態でも, 従来の15倍という広さで治療作業が行えることを示した. これらより, 狭い術野に向けて本システムをセッティングし, 精密で低侵襲な治療作業を実施できると考えられる.

ただし, 対象部位や症例によって, マニピュレータ部分の最適な仕様は異なる. そのため, 今後は特定の対象に合わせて, マニピュレータ部の仕様を変更し, 具体的な治療操作を検討することが必要である. そこで, 今後は動物を用いた臨床的な操作実験を行い, 本システムを外科医に評価いただく予定である.

7. 結論

患者の全身を対象に, 人の手や従来の術具, 手術用マニピュレータでは対応が困難な, 狭い術野での精密で低侵襲な手術の実現を目指して, 手術支援用マニピュレータシステムAMATERASを開発した.

本システムは, 脳神経外科分野で臨床応用に成功したHUMANシステムの基本機構を継承しつつ, マニピュレータの挿入姿勢を遠隔制御するためのポジショニングアームと, 八自由度の保持スタンドへのマニピュレータ搭載機構という二つの機構を新規開発した.

これらの機構により, 本システムで実現した主な機能の第1は, 患者へマニピュレータをセッティングできる姿勢と対象範囲の拡大である. これにより, 従来のHUMANシステムではセッティングできなかった身体部位での治療操作を可能にした.

第2は, 複数マニピュレータがアプローチ可能な治療可能領域の拡大である. マニピュレータとポジショニングアームの二重駆動システムを開発したことで, 従来のHUMANシステムの9倍(直径30mm)以上の範囲で治療可能となり, より多くの分野において, 精密で低侵襲な手術を実現する基本機構を開発した.

これらにより, 脳神経外科分野向けに実用化しているHUMANシステムの低侵襲手術支援技術を, より多くの分野を対象として, 人手や従来の手術用マニピュレータでは対応困難な, 狭隘な術野での手術に適用するための, 基本システムを開発した.

謝辞：本研究の一部は, 東京女子医科大学が新エネルギー・産業技術総合開発機構より受託した, 平成16年度産業技術研究助成事業の助成により実施した.

文献

1) Muhe E. Die erste Cholecystektomie durch das laparoskop. Langenbecks Arch Chir. 1986; 369: 804.
2) Shea J A, Healey M J, Berlin J A, et al. Mortality and Complications Associated with Laparoscopic Cholecystectomy. Ann Surg 1996; 224(5); 609-620.
3) 古川俊治, 若林剛, 小澤壮, 渡邊昌彦, 大上正裕, 北川雄光ら. Master-slave manipulatorを用いた手術と遠隔手術指導. 日本外科学会雑誌 2000; 101(3): 293-298.
4) Mohr F. W, Falk V, Diegeler A, et al. Computer-enhanced "ROBOTIC" Cardiac Surgery: Experience in 148 Patients. J Thorac. Cardiovasc Surg 2001; 121(5): 842-853.
5) Shennib H, Bastawisy A, McLoughlin J, Moll F. Robotic computer-assisted telemanipulation enhances

coronary artery bypass. J Thorac Cardiovasc Surg 1999; 117(2): 310-313.
6) Jinno M, Matsuhira N, Sunaoshi T, et al. Development of a Master Slave Combined Manipulator for Laparoscopic Surgery Functional Model and Its Evaluation. Proc MICCAI 2002, LNCS 2488. Tokyo, 2002: 52-59.
7) 西澤幸司, 河合俊和, 菅和俊, 藤江正克, 伊関洋, 高倉公朋ほか. 脳外科領域における手術用マニピュレータの臨床応用機の開発. 第42回日本エム・イー学会大会論文集 2003; 41(1): 453.
8) Goto T, Hongo K, Kakizawa Y, et al. . Clinical application of robotic telemanipulation system in neurosurgery. Case report. J Neurosurg 2003, 2003; 99: 1082-1084.
9) Hongo K, Goto T, Kakizawa Y, et al. Micro-manipulator system (NeuRobot) : clinical application in neurosurgery. Proc 17th Int Cong Exhib, CARS 2003. London, 2003: 509-513.
10) 西澤幸司, 村垣善浩, 伊関洋, 藤江正克, 佐久間一郎. MRI 連携手術支援マニピュレータシステムの開発. 第10回ロボティクスシンポジア予稿集 2005: 379-384.
11) 坪内広太, 原田香奈子, 千葉敏雄, 絵野沢伸, 藤江正克. 胎児手術用吸引型スタビライザの開発. JJSCAS 2005; 7(3): 315-316.

Paper:

Development of an Intraoperative Information Integration System and Implementation for Neurosurgery

Eisuke Aoki*, Masafumi Noguchi*, Jae-Sung Hong**, Etsuko Kobayashi***, Ryoichi Nakamura****, Takashi Maruyama****, Yoshihiro Muragaki****, Hiroshi Iseki****, and Ichiro Sakuma***

*Graduate School of Frontier Sciences, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: aoki@miki.pe.u-tokyo.ac.jp
**Department of Nanobiomedicine, Faculty of Medical Sciences, Kyushu University
3-1-1 Maidashi, Higashi-ku, Fukuoka 812-8582, Japan
***Department of Precision Engineering, School of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
****Institute of Advanced Biomedical Engineering and Science, Tokyo Women's Medical University
8-1 Kawada-cho, Shinjuku-ku, Tokyo 162-8666, Japan
[Received October 18, 2006; accepted March 24, 2007]

Complete resection of glioma is required to obtain a satisfactory outcome in neurosurgical treatment. It is difficult for neurosurgeons to identify the boundary between glioma and normal tissue using the naked eye alone, so surgical assistance systems such as surgical navigation systems for the detection of brain tumor have been used in clinical operations. Intraoperative information obtained from intraoperative biomedical measurement systems must be integrated to detect brain tumors more accurately. In this research, we developed an intraoperative information integration platform using middleware that has global positioning and global time management capabilities. To evaluate the platform, we developed an integrated platform consisting of devices and systems for neurosurgery. Through experiments, we confirmed the basic performance and effectiveness of our platform in a simulated clinical environment.

Keywords: system integration, neurosurgery, distributed system, 5-aminolevulinic acid, navigation system

1. Background

In recent years, a number of research laboratories are expected to develop devices and systems supporting minimally invasive surgery and its effects [1–3]. In order to perform more effective surgical operations, appropriate computer-assisted surgery is needed based on the required function, so an integrated system is needed that supports surgeons in an environment in which surgical devices are integrated.

In order to integrate a system, communication software is needed for connecting computers that control and manage multiple surgical devices, as is software to mutually integrate individual systems. If the development environments of individual systems differ, it takes much time and cost to develop such software.

As a solution, we have developed an intraoperative information integration system utilizing distributed object technology (middleware) [4]. When using distributed object technology, differences in operating systems (OSs) and hardware are made transparent and the development of application software operating under different platforms is facilitated. Parallel processing to distribute a load over multiple computers then becomes possible [5].

For these reasons, computer-assisted surgical systems using distributed object technology are being widely developed. Knappe et al., for example, developed a navigation robot reusing an existing system having a different development environment and specifications to reduce development time and cost [6]. Schorr et al. distributed the calculation load of image processing, which takes time in navigation processing, by using multiple PCs to develop a navigation system enabling real-time information presentation [7]. For improving performance by reducing development time and cost and distributing the load of calculation processing, computer-assisted surgical systems have thus been developed using distributed object technology [6–8].

When considering the construction of an integrated computer-assisted surgery environment, a problem arises in a distributed system using multiple computers because individual computers do not always indicate the same time, and the positional information of computer-assisted surgical devices may not be mutually associated correctly. To solve this problem, we must standardize time information, including mutual time synchronization of subsystem components and the integration of positional information. The integration of positional information has been fully discussed in an integrated system using multiple surgical devices [6–8]. Time has not, however, been sufficiently

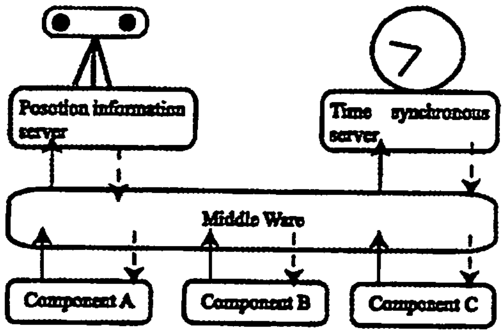

Fig. 1. System overview: system integration platform for surgery assisting system is consisted of global position server, global time server and components.

clarified and standardized in conventional studies.

In dealing with biological information, the positional information of organs varies from hour to hour as typified by bodily movement. To integrate biological information from multiple measuring instruments, the timing between measuring instruments must be standardized highly precisely.

Based on the above, we use not only distributed object technology and but also develop an integrated platform focusing on the integration of positioning and timing. Our discussions here focus on the highly precise standardization of timing, which is important for performing advanced computer-assisted surgery. We then apply the integrated platform to an actual surgery-assisted system for neurosurgery and verify its effectiveness through an evaluation experiment, as reported in the sections that follow.

2. Methods

2.1. System Configuration

We focus on three functions for realizing an integrated platform for a computer-assisted surgical system:

1. The integration of surgical devices having different development languages and environments.
2. The integration of positioning information.
3. The standardization of timing information.

In order to realize the integration of surgical devices having different development languages and environments, we use distributed object technology. In order to realize the integration of positioning information and the standardization of timing information, we established a time-synchronous server for providing standardized timing and positioning information servers for providing integrated positioning information in the integrated platform. We also provide communication software components for connecting components of surgical systems with the present platform (**Fig. 1**).

Table 1. Classification of middleware.

	OS independent	Language independent	Real Time
NDDS	○	○	◎
JAVA RMI	◎	×	△
COM	×	○	△
CORBA	○	◎	△

2.2. Use of Distributed Object Technology

Widely used middleware includes JAVA Remote Message Invocation (JAVA RMI), Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA), and the Network Data Distribution Service (NDDS) (**Table 1**). CORBA is actively studied and developed, having many supported development environments and programming languages [5–10].

Communication in CORBA is based on TCP/IP. In TCP, data lost during communication is retransmitted (Retransmission Control). Communication is controlled to adjust the amount of data based on available communication conditions (Congestion Control). TCP retransmission and congestion control is effective when data is transferred securely to a destination, but communication time cannot be estimated because data retransmission and the transmission amount are adjusted.

Unlike TCP, UDP does not perform retransmission control if communication information is lost during communication. While transmission of information is not ensured, communication having an immediate response is realized. In the control of mechatronics equipment, which requires strict real-time performance such as high-speed sampling, communication control having an immediate response rather than data reliability is needed. In controlling a computer-assisted surgery robot, UDP communication is preferable. By adding functions of retransmission and congestion control to the communication based on UDP, reliable communication such as TCP can be realized.

In the present study, we used NDDS (WaveWorks(R), RTI, USA), which is a type of distributed object technology based on UDP/IP [11]. NDDS uses a Real Time Subscribe (RTPS) communication model, which does not require an intermediation server or irrelevant request message.

2.3. Integration of Standard Positioning Information by Positioning Information Server

In managing systems by using an integrated coordinates, setting position of individual equipment are measured by using an external sensor, which we also use, and based on this, information from individual equipment is integrated into the standard coordinates.

External sensors include mechanical sensors measuring the location of the tip of a multijointed arm by an encoder detecting the angle of each joint, magnetic sensors detecting positioning by generating a magnetic field and measuring the intensity of the magnetic field by a magnetic field sensor attached to a probe, and optical sensors detecting positioning by using a CCD camera to capture the light reflected from an exclusive reflective marker attached to a probe.

The area measured by a mechanical sensor is limited to the driving range of a robot's mechanical arm. Magnetic sensor measurement error occurs due to magnetic materials around the target to be measured, and this restricts its use, depending on the surgical environment. Optical sensors involve the problem of blocking by obstacles, but enabling unrestricted measurement and comparatively higher measurement accuracy.

Based on the above, and considering reliability, expansion ease, precision, and the environment used in measurement, we used an optical three-dimensional positioning measurement device (POLARIS®, Northern Digital Inc. Canada) that uses reflective markers. Its sampling speed is approximately 60 Hz and root-mean-square (RMS) precision is 0.35 mm in positioning measurement of reflective markers. The positioning information server measures positioning information of the surgical device set up in the operating room and integrates coordinates with an optical position measurement device providing the standard coordinates.

2.4. Standardization of Timing Information by Time Synchronization Server

2.4.1. Requirement Specifications

OS generally standardize integrated time systems using network time protocols (NTP) via Internet connection. The NTP acquires correct time using the Global Positioning System(GPS) to adjust internal computer clocks based on world standard time.

Information must be managed highly confidentially when treatment information including private patient information used in surgery, which is why data is not usually transmitted or received via the Internet. Configuring practical computer-assisted surgery requires an integrated platform, regardless of the system environment, where information among computers in the operating room is limited to LANs (LANs). For this we use a time synchronization server generating a standard LAN time that differs from actual time.

To realize integrated computer-assisted surgery, information from multiple measuring instruments must be integrated so that subsystem time is synchronized with current standard time. In intraoperative logging recording surgery information, massive bleeding is a serious problem, before and after which information on equipment and operation is recorded as a time-series, allowing a lag from actual time.

The time synchronization server issuing standard time in an integrated system operates continuously. If a failure occurs in a single time synchronization server, standard time is lost, so the time synchronization server is rebooted, which loses consistency with time in intraoperative information recorded previously. Standard time in the integrated system must therefore be maintained regardless of problems in the time synchronization server.

When dealing with biological information, precise treatment information cannot be obtained if a time lag exists among measuring instruments. Tokuda et al. measured hepatic motion via breathing to show that a positioning variation of 50-60 mm is created at intervals of 7-8 seconds [12]. When position accuracy required for computer-assisted surgery is 1 mm, to obtain accurate treatment information for an ever-changing organism, time must be standardized to within 100 ms. To do so while considering the communication time lag, time resolution, and lag of measuring instruments and the time lag of the drive used together, it must be as fast as possible, so we set it to 30 ms. Under the system environment limited to the LAN, (a) time is precisely standardized at 30 ms and (b) the time synchronization server is fault-tolerant ensuring standard time continuity.

2.4.2. Time Synchronization Algorithm

A time synchronization server having high-precision standardization of time and fault-tolerant performance realizes functions according to the following flow:

(a) One of the subsystems is started as a time server.

(b) When a new system is added, time is synchronized with the present time server.

(c) If a failure occurs in the present time server, another subsystem becomes the time server to maintain the continuity of time management in the system.

2.4.3. High-Precision Standardization of Time

To realize high-precision standardization of time, it is necessary to consider two things:

- The lag created by the progress of the clock of each computer.
- The lag created by the communication time delay at the time of synchronization.

Regarding the lag created by the progress of the clock of each computer, progress can be adjusted by increasing the frequency of updating with the time synchronization server. The lag created by the communication time delay at the time of synchronization denotes the precision (hereafter, time synchronization precision) when synchronizing with the time synchronization server and the corrective effect of lags due to individual computer clocks depends on the time-synchronous precision of lags due to the communication time delay at time synchronization. Accordingly, to perform high-precision time synchronization, it is desirable to solve the problem of communication delay time.

In CORBA, TimeService is prepared as a service for synchronizing time [13]. It only obtains time issued by the TimeService, however, and communication delay time is not considered. In a one-to-one client-server system configured in a LAN environment, Dalton et al. show that there is an unsteadiness of communication time of from several ms to several tens of ms [14]. Schorr et al. show that a maximum 40 ms delay is created in communication processing time [7]. Depending on the communication environment to be used, reliable time synchronization is difficult to achieve when communication delay time is not considered.

In the present article, we employ a time-synchronous algorithm considering the communication time based on a formula (1) proposed by Cristian et al. [15]. In Cristian's algorithm (formula (1)), loopback time $(T2-T1-S)$ from the client to the server is measured, half of the loopback time $(T2-T1-S)$ is added to the time (G_{server}) sent from the server, and the time-synchronous time (G_{client}) is obtained.

$$G_{client} = G_{server} + \frac{(T2-T1-S)}{2} \quad . \; . \; . \; . \; . \quad (1)$$

G_{client}: time-synchronous time
G_{server}: time of time-synchronous server
$T2$: time when data is received from server
$T1$: time when data is sent to server
S: processing time at server

The problem with this method is that half of the loopback time cannot strictly be said to be communication time. The factor is cited that the round-trip communication time of the network is not always the same. When the loopback time is short enough, the round-trip time lag is small and can almost be neglected. That is, in the case of employing Cristian's method, time-synchronous precision is improved by synchronizing time when the loopback time is short, but communication conditions of the network are not fixed, therefore the decision criterion of whether loopback time is short or not should be decided according to circumstances.

In summary, in order to realize high-precision time synchronization, time synchronization is performed according to flows (a) to (c) as follows:

(a) Measure the loopback with the time server 100 times to calculate average (t) of the loopback time and its variation (σ).
(b) From a statistical value obtained from (a), communication time is estimated to be $(t-\sigma)$ when the network is relatively idle.
(c) When communication time is equal to or smaller than assumed value $(t-\sigma)$, synchronize time by adding communication time.

We adopt a policy in which, after time synchronization, if the consistency of time cannot be maintained among subsystems, for example, when transmitted time precedes reception time, synchronization is performed again.

2.4.4. Time Synchronization Server Having Fault Tolerance

The present system is designed to have failure tolerance using the following algorithm (Fig. 2):

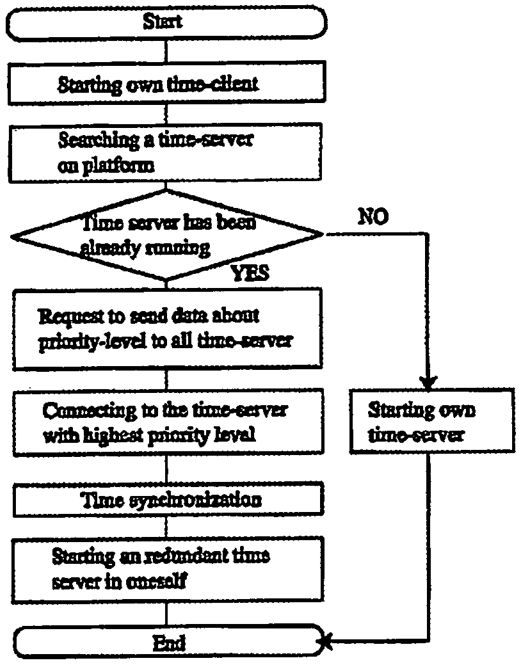

Fig. 2. Flow chart of fault tolerant time synchronization algorithm.

- All systems have time synchronization servers and clients communicating with the server to be connected to the platform.
- If no time synchronization server is present, the system itself becomes the time synchronization server.
- When a time synchronization server is present, the system becomes a client to be synchronized with the server. When multiple time synchronization servers are present, time is synchronized with the server having the highest priority.
- After completing time synchronization, the connected system drives the prioritized time synchronization server having priority. Clock performance is measured in advance and priority is allocated based on performance.

Thanks to such algorithms, even if the currently operating time synchronization server stops functioning, the server with second priority (the time synchronization server in operation having the highest priority) starts. The time consistency of the subsystem is maintained by the redundant time synchronization server (Fig. 3).

As mentioned before, actual time is not so important that prioritization based on the clock accuracy of the computer does not have an important meaning. Changing to a time synchronization server having a largely different clock performance, however, is not desirable because there is the possibility that a time lag may be created in the integrated system. Clock performance is therefore measured in advance and priority is allocated in turn based on performance.

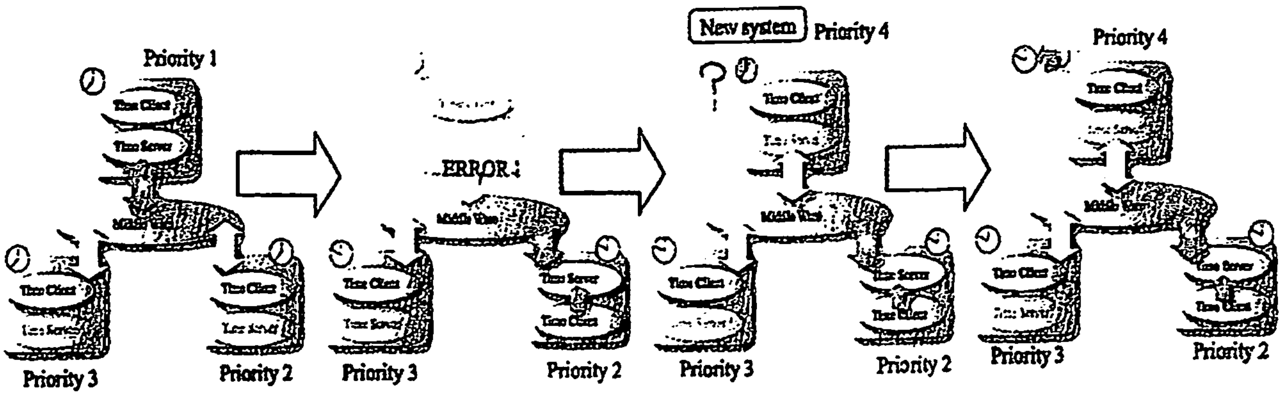

Fig. 3. Fault tolerant algorithm of the whole system; priority 1 > 2 > 3 > 4.

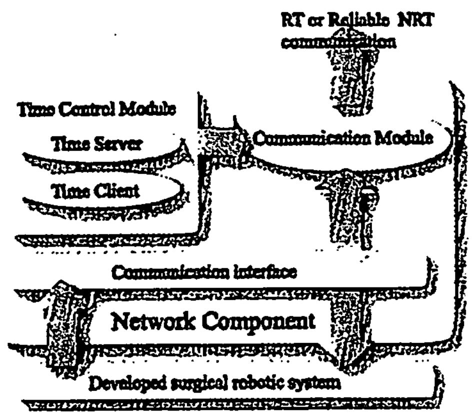

Fig. 4. Overview of network component.

2.5. Communication Component Module

To realize a time synchronization server having time consistency and fault-tolerant performance, as mentioned above, software to perform the function is required to be implemented for each component connected to the platform. It is problematic, however, in view of development efficiency, to have separate correspondence to software implementation operation for each system, we developed a communication component (class library) having the time synchronization function above (Fig. 4).

Depending on the computer-assisted surgical system to be used, the requirement for communication may differ. When controlling a computer-assisted surgery robot, for example, communication having an immediate response is required. When recording intraoperative information for which security is important, an immediate response is not necessary; instead, ordered and secure transmission and reception of information is required.

We therefore implement two types of communication modules into the communication component to enable communication to be selected based on the system: a RealTime communication module (RT communication) to realize communication having an immediate response based on UDP/IP and a Reliable communication module (Reliable communication) to make data transmission and reception secure through the functions of retransmission control and congestion control with no real-time performance. When information to be transmitted is being transferred from a corresponding surgical device, the communication component starts a thread performing time synchronization in the background to synchronize time with the time synchronization server in the network. When data is transferred through a communication interface from the corresponding surgical device, integrated time information is added to the data as a time stamp and data is transmitted and received by a selected communication module (RT communication or Reliable communication). Regarding such a time synchronization function and communication control of the network, all processing is performed in the background.

The above points ensure data transfer between platforms for users without having to change the control software of the existing surgical device, etc., simply by deciding data interface transmission to and reception from the network.

3. Application to Neurosurgical Surgery-Assisted System

We have associated the fluorescence intensity and spectrum information obtained from 5-aminolevulinic acid (5ALA) induced protoporphyrin IX (Pp9) fluorescence with cancer malignancy as a neurosurgical surgery-assisted system to develop a cancer detection system for quantitative diagnosis [1,2]. Even if a cancerous area is identified by this cancer detection system, it is difficult to remove cancer if it is close to an important functional region but the positional relationship of a patient's motor language area to the motor area (functional information) is not sufficiently clear.

In surgery using a navigation system based on preoperative diagnostic information, the location of the unclear cancer boundary (anatomical information) and the cerebral function region (functional information) of motor language area and motor area, etc., is presented to the surgeon as objective three-dimensional image information. More effective surgical navigation is realized with histological information such as cancer malignancy detected by Pp9 fluorescence measurement being presented to the surgeon together with cerebral functional and anatomic information by this navigation system.

We therefore applied the position server, time server, and communication component we developed to a surgery-assisted system for neurosurgery. We configured an integrated navigation system that collects and integrates information from multiple intraoperative measuring instruments to display on a navigation screen. Functions of the integrated system are realized in the following flow:

(a) Standardization of time between subsystems.

(b) Integration of coordinates to unify positioning-information by using a positioning-information server.

(c) Collection of multiple intraoperative information associated with positioning.

(d) Integration of information from time information added simultaneously to intraoperative information.

(e) Integration of preoperative and intraoperative information obtained from a diagnostic imaging system.

(f) Extraction of integrated information based on surgical conditions to display as intuitive information on the navigation screen to the surgeon.

The integrated system consists of the following five computer-assisted surgical systems.

3.1. Cancer Detection System by Intraoperative Cerebral Cancer Fluorescence Diagnosis Using 5ALA-Induced PpIX

5-ALA-induced fluorescent material Pp9 in cancer tissue is excited locally using a 405 nm excitation laser to detect the red fluorescence of an emission wavelength peak of 635 nm using a detector made of optical fiber. The spatial resolution of the detector is 0.6 mm. Detected fluorescence is spectrally analyzed by a spectral photometer, then histological information on the malignancy and type of cancer is obtained as quantitative information from the intensity and spectrum of the fluorescence signal (Fig. 5(b)) [1].

3.2. Autofocusing Robot

It is essential to stable fluorescence measurement to keep sufficient distance between the fluorescence detector and the subject being measured. This requires an automated positioning control function that unfailingly makes the operating distance between the fluorescence detector and the object surface coincide. In the present study, we placed a fluorescence probe on a surgical microscope (Mitaka Kohki Co., Ltd.) equipped with an automated positioning control function that maintains a fixed distance between the microscope and the object being measured

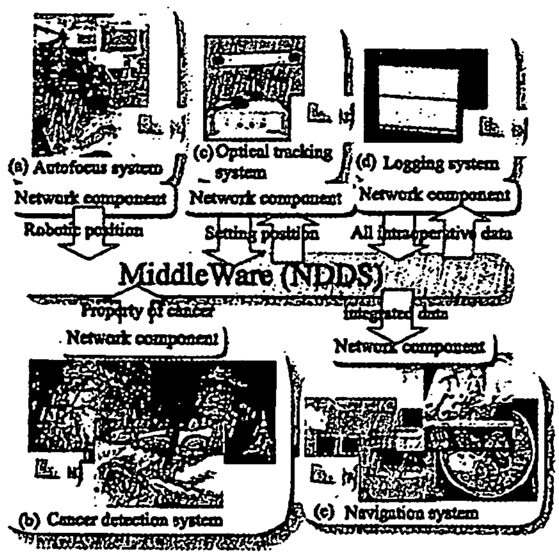

Fig. 5. Integrated neurosurgical robotic platform using NDDS.

(Fig. 5(a)). By combining a two-axis robot stage (scanning robot), positioning measurement becomes possible while the robot drive is scanning the brain surface.

In the present system, we measure positioning using a confocal optical system in a confocal laser microscope. Positioning measurement, which distinguishes top and bottom of an object using a principle similar to the optical pickup of CDs, becomes possible by concentrating a guide laser with a wavelength of 670 nm on the surface of an object and using a two-fraction diode in place of a pin hole in the detector [2]. When the object is a metal with stain finished surface, it is possible to ensure positioning with an accuracy of 2.0 μm.

3.3. Navigation System

Three-dimensional image information is prepared based on preoperative diagnosis information such as Magnetic Resonance Imaging (MRI), and spatial positioning of the object being treated is intuitively shown to the surgeon. As the navigation system, we used a system developed by the Kyushu University Center for the Integration of Advanced Medicine and Innovative Technology that is based on a 3D slicer [16], which is free software. The navigation system [3] displays three-dimensional imaging information, including the boundary between cancerous and normal tissue and functional information on the brain, etc., to the surgeon qualitatively and in real time (Fig. 5(e)).

3.4. Positioning-Information Server (Section 2.3)

As mentioned in Section 2.3, by using an optical position measurement device as the position measurement device, three-dimensional positioning-information of the surgical device established in the operating room is measured and coordination integration is performed (Fig. 5(c)).

3.5. Intraoperative Logging System

An intraoperative logging system monitors and records intraoperative logging information such as the condition and function of measuring instruments during surgery. By tracing and analyzing logging information, it is determined whether the surgical robot and measuring instruments are functioning without problems, and if problems occur, the cause of the error is checked (Fig. 5(d)).

4. Evaluation Experiments

We confirmed high-precision time standardization of 30 ms and the fault-tolerant performance of the time-synchronous server through evaluation experiments in a laboratory environment. We also confirmed functions of the entire integrated system through an *in vivo* experiment simulating clinical use.

4.1. Evaluation of Integrity of Integrated Time

In cerebral nerve surgery, some surgeries take as long as 10 hours or more to remove malignant cancer.

Assuming that the computer-assisted surgical system is applied practically in clinical use, we evaluated the integrity of time information with and without time-synchronization during 12 hours of continuous system operation.

4.1.1. Experimental Methods

For experiments, we prepared intraoperative logging (OS: Redhat 9.0, CPU: 3.2 GHz, Memory: 1024 MB) in which the time-synchronous server is operated, measurement system A (OS: Windows XP, CPU: 3.04 GHz, Memory: 1024 MB) using Windows, and measurement system B (Redhat 9.0, CPU: 3.04 GHz, Memory: 1024 MB) using Linux. After ensuring time synchronization once, the current time of each measurement system is transmitted to intraoperative logging at intervals of 20 ms. The time of each system received in a 20 ms cycle is recorded along with the time of the time-synchronous server at intervals of 1 s.

Under such circumstances, we evaluated time lag with the time-synchronous server with the system as is for 12 hours when (a) time is synchronized every 20 minutes and (b) time synchronization is not done and left as is. The time lag evaluated is the difference between the time of each measurement system, which intra-operative logging receives, and the time when the time-synchronous server is recorded. Since measurement and intra-operative logging ensure asynchronous communication, 20 ms becomes the time resolution, which is evaluated and equivalent to the transmission period. Communication is in a 100 Mbase-T LAN environment.

4.1.2. Results of Experiments

Table 2 shows time lag with the time-synchronous server using the elapse of one hour as a criterion when time synchronization is not done and when the time is synchronized every 20 minutes for Windows and Linux. When time is not synchronized, an increase in the time lag is seen over time. The time until 30 ms, a required specification we set, is exceeded takes 8 minutes for Windows and 26 minutes for Linux.

Table 2. Time lag with and without a timeserver.

Elapsed time [hour]	Windows without timeserver [msec]	Windows with timeserver [msec]	Linux without timeserver [msec]	Linux with timeserver [msec]
1	265	97	69	10
2	610	41	130	11
3	1016	41	184	11
4	1392	41	232	10
5	1848	42	289	11
6	2265	42	341	11
7	2265	43	341	11
8	3093	43	445	10
9	3502	43	498	10
10	3915	44	550	11
11	4330	43	604	10
12	4743	28	657	11

When time is synchronized every 20 minutes, no increase in time lag was seen over for either Windows or Linux. More clock fluctuation was seen for Windows than Linux, i.e., when the time lag is 10-11 ms for Linux, it is 28-97 ms for Windows. In the experiment below, time-synchronous server priority is preferentially given to Linux due to its smaller clock fluctuation. For the same OS, priority is given to the computer having better performance. To realize time synchronization of 30 ms, time synchronization frequency is given a margin such as every 20 minutes for Linux and every 5 minutes for Windows.

4.2. Evaluation of Time Synchronization Precision

Time synchronization precision is evaluated by time synchronization considering the communication delay time mentioned in Section 2.4.3.

4.2.1. Experimental Methods

The integrated system used in the experiment consists of the following components: a navigation system (OS: Redhat 9.0, CPU 3.2 GHz, Memory: 2048 MB), in which time-synchronous server is operated; an intraoperative logging system (OS: Redhat 9.0, CPU 3.2 GHz, Memory: 2048 MB); a cancer identification system (OS: Windows XP, CPU: 3.2 GHz, Memory: 2048 MB); a positioning-information server (OS: Windows XP, CPU:3.2 GHz, Memory: 2048 MB); and an autofocusing robot (OS: Windows XP, CPU: 2.8 GHz, Memory: 512 MB). The communication condition is a 100 Mbase-T LAN environment. Each system performs transmission and reception depending on the communication conditions shown in Table 3.

Under the condition that only the navigation system be first connected to the integrated system, the intraoperative

Table 3. Communication condition (100 Mbase-T).

Agent	Communication data [byte]	Sampling Frequency [Hz]
(a)CancerDetection	1080	5
(b)AutoFocusingRobot	243	20
(c)Registration	942	10
(d)Navigation	1212	5
(e)Logging	2265	50

Table 4. Result of time synchronization (n=100).

Agent	Average of loop back time ($T2-T1$) [msec]	Average of processing time (S) [msec]	Max loop back time [msec]
Logging	1.7±0.2	0.01	2.32
CancerDetection	0.59±0.13	0.03	1.69
AutoFocusingRobot	0.66±0.7	0.02	0.87
Registration	1.05±0.4	0.02	1.37

logging system, cancer identification system, positioning-information server, and autofocusing robot are connected to the platform sequentially turn to perform time synchronization with the intraoperative logging system.

4.2.2. Results of Experiments

As shown in Table 4, the time synchronization precision of each measurement system is within 2 ms and the maximum delay time is 2.3 ms.

4.3. Evaluation of Failure Tolerance of Time-Synchronous Server

We confirmed that if problems occur in the time-synchronous server, a substitute time-synchronous server is started and time consistency in the integrated system is maintained.

4.3.1. Experimental Methods

The integrated system used in the experiment consists of the following components: an autofocusing robot (OS: Windows, CPU 3.2 GHz, Memory: 2048 MB) (A); a cancer identification system (OS: Windows, CPU 3.04 GHz, Memory: 2048 MB) (B); a positioning-information server (OS: Windows, CPU 2.8 GHz, Memory: 512 MB) (C); and an intraoperative logging system (OS: Windows, CPU 1.04 GHz, Memory: 1024 MB). The system time of each system is recorded by the intraoperative logging system. The communication condition is a 100 Mbase-T LAN environment. Time-synchronous servers are mounted in (A), (B), and (C). Priorities of servers are set as (A)>(B)>(C). The time-synchronous server is first operated in (A).

Under the condition that systems (A), (B), and (C) are already connected to the integrated system, (A) and the time-synchronous server operating in (A) are disconnected from the integrated system. In a short time, (A)

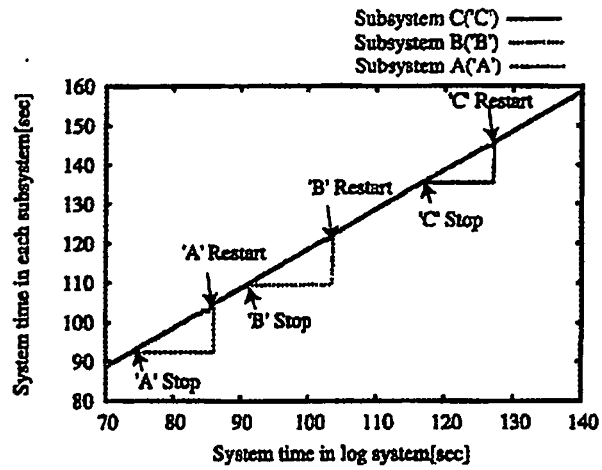

Fig. 6. Uniformity of time according to stop and restart of each timeserver.

is reconnected with the integrated system. We confirmed that even when (A) is reconnected, by synchronizing with the time-synchronous server (B), which started as a substitute for the (A), the time of (A) is made to match the time in the integrated system to maintain time consistency. For (B) and (C), it is performed under the same experimental conditions.

4.3.2. Results of Experiments

Figure 6 shows results of the measurement of time of each system accompanied by disconnection and reconnection of the system. The horizontal axis shows the system time for the intraoperative logging system and the vertical axis shows the integrated system time of systems (A), (B), and (C). As shown by Fig. 6, after disconnecting (A) from the integrated system, (A) is reconnected to the integrated system after an interval of several seconds. Compared to the time of (B) and (C), the time of (A) when reconnected has a time lag of 0 ms from (B) and 24 ms from (C), and integrated into the time in the integrated system without a lag of 30 ms or more. (A) performed time synchronization with the time server (B), which started as the substitute server for (A) in the integrated system. We confirmed that the time-synchronous server is correctly selected based on the specified priority.

Similarly, we confirmed that when disconnecting and reconnecting from the integrated system, the time of systems (B) and (C) is integrated with the time in the integrated system without a time lag of 30 ms or more.

4.4. Evaluation of Entire System by *in Vivo* Experiment

We confirmed basic navigation function under an environment simulating clinical use.

The integrated system is configured using five separately developed computer-assisted surgery systems (Fig. 5) (Section 3.1) communicating with each other. Communication interfacing of each system is defined as Table 5. Communication conditions are set as shown in

Table 5. Interface definition.

Agent	Communication interface
(a)CancerDetection	Property of cancer, measurement time
(b)AutoFocusingRobot	Robotic position, measurement time
(c)Registration	Setting position, measurement time
(d)Navigation	Integrated data(robotic position, setting position and property of cancer)
(e)Logging	All intraoperative data

Table 6. Computer specification.

Agent	RAM [MB]	CPU [MHz]	OS
(a)CancerDetection	512	2800	Windows XP
(b)AutoFocusingRobot	1024	3060	RedHat 9.0
(c)Registration	1024	3060	RedHat 9.0
(d)Navigation	1024	2400	FedoraCore 3.0
(e)Logging	2048	3060	Redhat 9.0

Table 3. Communication is done under the 100 Mbase-T LAN environment for the number of clients and (Table 3) communication data used in the integrated system, which ensures sufficient communication resources. Table 6 shows specifications for computers of computer-assisted surgical devices. Based on specifications in Table 6 and the policy under Results Section 4.1, time-synchronous server priority is set to (e), (c), (b), (d), and (a), in this sequence. Originally, data displayed in navigation is MRI images captured before surgery, but because preoperative image by MRI cannot be obtained in animal experiment facilities, dummy data is used, consisting of a porcine head imaged in advance.

4.4.1. Experimental Methods

An anesthetized pig (Landrace) is subjected to a craniotomy and 50 mg/kg of 5ALA, which is an excess quantity, is administered intravenously. In human subjects orally administered 5ALA, in the process of metabolism, 5ALA changes into PpIX, becoming fluorescent that accumulates selectively only in cancerous tissue. In our experiments, by intravenously administering greater amounts than those used in ordinary surgery, both cancerous and normal tissue uptakes the dye, 5ALA inducing PpIX, which accumulates and fluoresces from all exposed brain surfaces, where body movement of 5-10 mm is created at intervals of 6-7 seconds caused by the pig's breathing and heartbeat.

Based on such an *in vivo* test environment, we intentionally prepared nonfluorescent regions (Fig. 7(a) above), where the brain is covered by dura matter, protecting the brain surface, and fluorescent regions (Fig. 7(a) below), where the brain surface is exposed. Fig. 7(b) shows the brain surface exposed in reverse imaging and the unexposed region in normal imaging. Using autofocusing and scanning robots, scanning spans the boundary between fluorescent and nonfluorescent regions (Fig. 7) while maintaining a fixed distance between the instrument and object. Pp9 fluorescence is measured by a fluorescence detector having a spatial resolution of 0.6 mm.

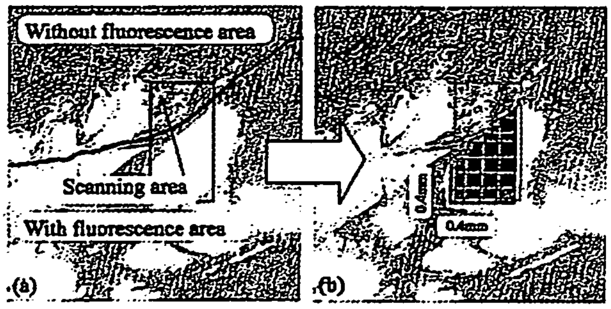

Fig. 7. Overview of brain surface with and without fluorescence area.

Table 7. Result of time synchronization (n=100).

Agent	Average of loop back time ($T2-T1$) [msec]	Average of processing time (S) [msec]
Logging	1.16±0.05	0.25
CancerDetection	0.7±0.28	0.21
AutoFocusingRobot	0.7±0.3	0.34
Registration	1.0±0.4	0.29

Specifically, a selected region is divided (Fig. 7(b)) into lattices at intervals of 0.4 mm, scanning stops on each lattice intersection and fluorescence is measured, then the next lattice point is moved to and measured. When fluorescence is measured, the spectral photometer is set to an exposure time of 200 ms. Fluorescence information (Fig. 7(b)) from cancer identification and information from the autofocusing robot showing the measurement position are integrated based on integrated time and displayed on the navigation screen as region information on fluorescent (red) and nonfluorescent (white) region.

4.4.2. Results of Experiments

Table 7 shows results of measurement of the time lag between the time-synchronous server of each system. Integrated system time is integrated with time-synchronous precision of 1 ms. After we conducted experiments for two hours, the time lag between the intra-operative measurement system and time-synchronous server was within 30 ms, i.e., it is 26 ms for the intra-operative measurement system (Windows) and 11 ms for the auto-focusing robot (Linux).

Figure 8 shows results of superposing the actual brain surface boundary of the region covered by dura matter (nonfluorescent) and that (fluorescent, reverse image) where the brain surface is exposed and the boundary (red: fluorescent, white: nonfluorescent) displayed on the navigation screen.

Unlike the MRI image of the pig, information measured during surgery is obtained as fluorescent by ensuring information integration based on the integrated time and displayed on the navigation screen having preoperatively photographed MRI image information, indicating that basic navigation functions are realized.

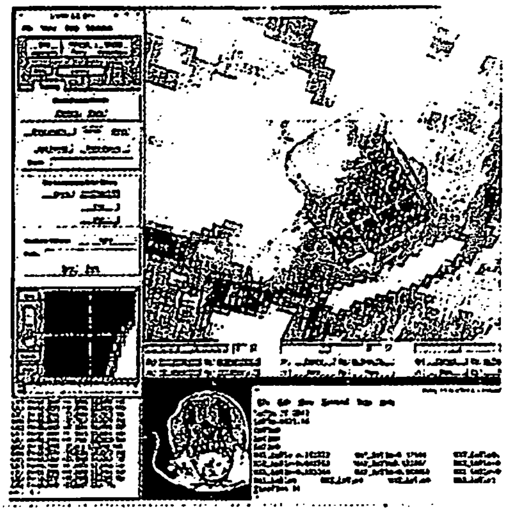

Fig. 8. Integration of Pp9 Spectrum data and 3D position data.

5. Discussion

5.1. Evaluation of Integrity of Integrated Time

Surgical information often is too sensitive, e.g., private patient information, to be sent via the Internet connection and NTP is often not available. When configuring a distributed system under such circumstances, a time lag exists between computer clock time and actual time. Given that integrated systems into surgery are continuously used for at least two to three hours, experiments indicate a time lag of about one second is created when using Windows.

As clarified by the result of the experiment, to realize time-synchronous precision of 30 ms required in our work, time synchronization must be within 8 minutes for Windows and 26 minutes for Linux. Since the clock fluctuates with Windows, we concluded that the frequency of the time-synchronous server must have a margin to realize time synchronization every 20 minutes for Linux and every 5 minutes for Windows.

5.2. Evaluation of Time Synchronization Precision

One may have an objection that if there is sufficient real time performance, when each time is different, there is no obstacle to function. Even if real-time performance of individual equipment software is ensured, it is unclear whether real-time performance is ensured for communication between equipment. Even if sufficient communication resources are ensured in the LAN when using the network, a communication lag of a maximum 40 ms may arise [7]. In the liver, for example, the positioning fluctuation of 50 to 60 mm may exist at intervals of 7 to 8 seconds. A time lag between measuring instruments could make treatment information unreliable when information is integrated. Our proposed time synchronization realizes precise time synchronization by considering communication time lags by measuring loop-back time with the time synchronization server. After determining network communication conditions, time is synchronized using Cristian's algorithm when a communication time lag is sufficiently short. Our experiments showed that time synchronization with precision of 2 ms is feasible.

5.3. Evaluation of Failure Tolerance of Time-Synchronous Server

Since sufficient time precision is maintained for short periods when switching to the clock in each computer, it could be assumed that fault-tolerant performance already existed.

Even when the time-synchronous server is disconnected, the time lag of each computer is considered small when system operating time is short. As discussed in Section 5.1.1, a time lag of 30 ms exists for 8 minutes for Windows. When considering that computer-assisted surgery is operates continuously for two to three hours, it becomes difficult to maintain time consistency in the integrated system without the time-synchronous server. As mentioned in Section 2.4.1, when a standard time-synchronous server is disconnected from the integrated system, consistency of information cannot be maintained.

The fault-tolerant time-synchronous server we propose copes with these problems because even if problems occur in the time-synchronous server, it continues to provide standard time in the integrated system.

5.4. Evaluation of Entire System by *in Vivo* Experiment

To reduce cost and development time when changing specifications and adding functions, we used distributed-object technology, developed communication components in which all functions such as time synchronization are involved, and applied these to a neurosurgery-assisted system.

An integrated navigation system is configured by integrating already developed computer-assisted surgical devices, but it takes almost a week. We have no choice but to evaluate development time with qualitative and subjective indexes because communication software is developed separately based on the environment is no longer required. Since time consistency is realized in the background by the communication component we developed, labor for developing the integrated system is reduced.

Through an *in vivo* evaluation experiment, we standardized time, which is important for integrating intra-operative information, using time-synchronous precision of 1 ms, equivalent to results of experiments in Section 4.2, i.e., 2 ms. The time lag is within 30 ms under continuous 2-hour system operation. Basic functions of integrated navigation are confirmed and that intra-operative measurement information from measurement systems such as cancer identification and the auto-focusing robot is integrated based on standardized time to be displayed on a navigation screen having preoperative diagnosis information.

Positron emission tomography/computed tomography (PET/CT) is an example ensuring multiple information