

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4505606号
(P4505606)

(45) 発行日 平成22年7月21日(2010.7.21)

(24) 登録日 平成22年5月14日(2010.5.14)

(51) Int.Cl.		F I			
H04L 12/56	(2006.01)	H04L 12/56	100Z		
H04L 12/28	(2006.01)	H04L 12/28	100F		
B25J 19/00	(2006.01)	B25J 19/00	J		

請求項の数 9 (全 38 頁)

<p>(21) 出願番号 特願2005-103582 (P2005-103582)</p> <p>(22) 出願日 平成17年3月31日(2005.3.31)</p> <p>(65) 公開番号 特開2006-287520 (P2006-287520A)</p> <p>(43) 公開日 平成18年10月19日(2006.10.19)</p> <p>審査請求日 平成20年3月3日(2008.3.3)</p> <p>特許法第30条第1項適用 平成16年10月25日 情報処理学会関西支部主催の「平成16年度 情報処理 学会関西支部 支部大会」において文書をもって発表</p> <p>特許法第30条第1項適用 2004年12月17日 社団法人計測自動制御学会発行の「第5回 社団法人計 測自動制御学会 システムインテグレーション部門 講 演会 S12004 講演概要集」に発表</p> <p>特許権者において、実施許諾の用意がある。</p>	<p>(73) 特許権者 393031586 株式会社国際電気通信基礎技術研究所 京都府相楽郡精華町光台二丁目2番地2</p> <p>(74) 代理人 100090181 弁理士 山田 義人</p> <p>(72) 発明者 宮下 敬宏 京都府相楽郡精華町光台二丁目2番地2 株式会社国際電気通信基礎技術研究所内</p> <p>(72) 発明者 石黒 浩 京都府相楽郡精華町光台二丁目2番地2 株式会社国際電気通信基礎技術研究所内</p> <p>審査官 吉田 隆之</p> <p style="text-align: right;">最終頁に続く</p>
--	--

(54) 【発明の名称】 皮膚センサネットワーク

(57) 【特許請求の範囲】

【請求項1】

少なくとも1つの触覚センサをそれぞれが備えかつ少なくとも1つの他のノードと互いに接続された複数のノードと、少なくとも1つのノードと接続されたホストコンピュータとを含む皮膚センサネットワークであって、

前記ホストコンピュータは、

接続された前記ノードに経路要求を送信する第1経路要求送信手段、

前記経路要求に応じて送信される経路情報を前記接続されたノードから受信して第1記憶手段に記憶する経路情報取得手段、

前記経路情報に基づいて前記複数のノードのそれぞれに対する触覚情報要求を前記接続されたノードに送信する触覚情報要求送信手段、および

前記触覚情報要求に応じて送信される触覚情報を前記接続されたノードから受信して前記第1記憶手段に記憶する触覚情報取得手段を含み、

前記複数のノードのそれぞれは、

前記触覚センサの出力データを検出して第2記憶手段に記憶するセンサ出力検出手段

、

前記経路要求を受信したとき経路確保するか否かを判定する経路確保判定手段、

前記経路確保判定手段によって経路確保すると判定されたとき当該経路要求を受信したポートをホスト側経路として前記第2記憶手段に記憶する経路記憶手段、

前記経路確保判定手段によって経路確保すると判定されたとき承諾応答を前記ホスト

10

20

側経路に送信する応答送信手段、

前記承諾応答を受信したとき当該承諾応答を受信したポートに経路確保確認を送信する確認送信手段、

前記経路確保確認を受信したとき前記ホスト側経路以外に前記経路要求を送信する第2経路要求送信手段、

前記第2経路要求送信手段によって送信した前記経路要求に対する承諾応答がないと判定されるとき、自分の経路情報を前記ホスト側経路に送信する第1経路情報送信手段、

前記経路情報を受信したことに応じて、自分の経路情報に受信した経路情報を結合することによって生成した経路情報を前記ホスト側経路に送信する第2経路情報送信手段、

前記触覚情報要求を受信したことに応じて、前記センサ出力検出手段によって検出された前記出力データを触覚情報として前記ホスト側経路に送信する第1触覚情報送信手段、および

前記触覚情報を受信したとき当該触覚情報を前記ホスト側経路に送信する第2触覚情報送信手段を含む、皮膚センサネットワーク。

【請求項2】

前記各ノードは、前記センサ出力検出手段によって検出された前記出力データを圧縮する圧縮手段をさらに含み、

前記第1触覚情報送信手段は、前記触覚情報要求を受信したことに応じて、前記圧縮手段によって圧縮されたデータを前記触覚情報として前記ホスト側経路に送信する、請求項1記載の皮膚センサネットワーク。

【請求項3】

前記各ノードは、

前記センサ出力検出手段で検出した前記触覚センサの前記出力データを当該触覚センサの隣接センサの所属するノード宛に送信する隣接センサ出力送信手段、および

前記隣接センサの所属するノードからの当該隣接センサの出力データを受信して前記第2記憶手段に記憶する隣接センサ出力取得手段をさらに含み、

前記圧縮手段は、前記触覚センサの前記出力データをマスクするか否かを、前記センサ出力検出手段によって検出されたまたは前記隣接センサ出力取得手段によって記憶された当該触覚センサの隣接センサの出力データに基づいて判定する第1マスク判定手段を含み、

前記第1触覚情報送信手段は、前記第1マスク判定手段によってマスクしないと判定された前記触覚センサの前記出力データを、前記触覚情報として送信する、請求項2記載の皮膚センサネットワーク。

【請求項4】

前記第1マスク判定手段は、前記隣接センサの出力値の平均を中心とするかつ前記出力値の差に第1係数を掛けた値を幅とする第1マスク領域を算出するマスク領域算出手段を含み、前記触覚センサの出力値が前記第1マスク領域内にあるとき、当該触覚センサの出力データをマスクすると判定し、前記第1マスク領域内にないときは当該出力データをマスクしないと判定する、請求項3記載の皮膚センサネットワーク。

【請求項5】

前記第1マスク判定手段は、前記隣接センサの出力値に対する前記触覚センサの出力値の変化量が第1閾値よりも小さいとき、当該触覚センサの出力データをマスクすると判定する第1ノイズマスク判定手段をさらに含む、請求項4記載の皮膚センサネットワーク。

【請求項6】

前記圧縮手段は、前記触覚センサの前記出力データをマスクするか否かを、前記センサ出力検出手段によって検出された当該触覚センサの出力データの履歴に基づいて判定する第2マスク判定手段を含み、

前記第1触覚情報送信手段は、前記第2マスク判定手段によってマスクしないと判定された前記触覚センサの前記出力データを、前記触覚情報として送信する、請求項2ないし5のいずれかに記載の皮膚センサネットワーク。

10

20

30

40

50

【請求項 7】

前記第 2 マスク判定手段は、前記触覚センサの過去の出力値からの変化量が第 2 閾値よりも小さいとき、当該触覚センサの出力データをマスクすると判定する第 2 ノイズマスク判定手段をさらに含む、請求項 6 記載の皮膚センサネットワーク。

【請求項 8】

前記各ノードは、前記触覚情報を受信したとき送信データの優先順位を判定する優先順位判定手段をさらに含む、

前記第 2 触覚情報送信手段は、前記優先順位判定手段によって受信した前記触覚情報の優先順位が高いと判定されたとき、当該受信した触覚情報を優先的に送信する、請求項 1 ないし 7 のいずれかに記載の皮膚センサネットワーク。

10

【請求項 9】

少なくとも 1 つの触覚センサをそれぞれが備えかつ少なくとも 1 つの他のノードと互いに接続された複数のノードと、少なくとも 1 つのノードと接続されたホストコンピュータとを含む皮膚センサネットワークであって、

前記ホストコンピュータは、

接続された前記ノードに経路要求を送信する第 1 経路要求送信手段、および

前記経路要求に応じて送信される触覚情報を含んだ経路情報を前記接続されたノードから受信して第 1 記憶手段に記憶する情報取得手段を含み、

前記複数のノードのそれぞれは、

前記触覚センサの出力データを検出して第 2 記憶手段に記憶するセンサ出力検出手段

20

、
前記センサ出力検出手段によって検出された前記出力データを圧縮する圧縮手段、
前記経路要求を受信したとき経路確保するか否かを判定する経路確保判定手段、
前記経路確保判定手段によって経路確保すると判定されるとき当該経路要求を受信したポートをホスト側経路として前記第 2 記憶手段に記憶する経路記憶手段、

前記経路確保判定手段によって経路確保すると判定されるとき承諾応答を前記ホスト側経路に送信する応答送信手段、

前記承諾応答を受信したとき当該承諾応答を受信したポートに経路確保確認を送信する確認送信手段、

前記経路確保確認を受信したとき前記ホスト側経路以外に前記経路要求を送信する第 2 経路要求送信手段、

30

前記第 2 経路要求送信手段によって送信した前記経路要求に対する承諾応答がないと判定されるとき、前記圧縮手段による圧縮に基づく触覚情報を含んだ自分の経路情報を前記ホスト側経路に送信する第 1 情報送信手段、および

前記経路情報を受信したことに応じて、前記圧縮手段による圧縮に基づく触覚情報を含んだ自分の経路情報に受信した前記経路情報を結合することによって生成した、触覚情報を含んだ経路情報を前記ホスト側経路に送信する第 2 情報送信手段を含む、皮膚センサネットワーク。

【発明の詳細な説明】

【技術分野】

40

【0001】

この発明は、皮膚センサネットワークに関し、特にたとえば、触覚センサエレメントを備える複数のノードとホストコンピュータとを含む皮膚センサネットワークに関する。

【背景技術】

【0002】

人間と共存しコミュニケーションするロボットが近年開発されてきており、ロボットの活動範囲も人間の近くになりつつある。人間と日常生活環境で共存するロボットにとって、皮膚感覚、特に触覚は重要な感覚の 1 つとなる。動物においては感受性の主となるものが触覚であるといわれており、ロボットも全身分布型の触覚を通して触りまたは触られることができれば、ロボットと人間とのインタラクションがより豊富なものになることが期

50

待される。

【 0 0 0 3 】

これまでロボットの触覚に関しては、マニピュレータやロボットハンドの把持・操りのためのタスクに関する研究は多いが、ロボットと人間のインタラクションに必要となってくる全身触覚に関する研究は比較的少ない。高密度の分布型触覚デバイスで様々な形状のロボットの体表を覆い、大量のセンサ情報をホストコンピュータに送信して処理する場合には、物理的な配線問題や通信経路・通信速度などのセンサ - ホスト間通信に関する様々な問題があるためであると考えられる。なお、このように通信機能をもつ多数のセンサを配置して環境の観測や設備の管理等に役立てるためのシステムは、センサネットワークと呼ばれる。

10

【 0 0 0 4 】

たとえばロボットの全身触覚に関しては、ロボットの体表に実装できる触覚デバイス（非特許文献1）や、触覚情報を送る手法（非特許文献2）などが研究されている。具体的には、非特許文献1の技術は、多値接触センサを読み取ってホストコンピュータに情報を送るノードがバス接続されたシステムである。また、非特許文献2の技術では、MEMS（Micro Electro Mechanical System）による小型接触センサと無線ユニットとを一体化した小型チップをロボット表面に分布させることで階層化された無線ネットワークの形成を可能にしている。

【非特許文献1】陰山竜介（外4名）、“ロボット表面多値接触センサの開発と応用”、ロボティクスメカトロニクス講演会'98講演論文集（1C11-2）

20

【非特許文献2】Hiroyuki Shinoda（外3名）、“Two-Dimensional Signal Transmission Technology for Robotics”，Proceedings of the 2003 IEEE International Conference on Robotics & Automation Taipei, Taiwan, September 14-19, 2003

【発明の開示】

【発明が解決しようとする課題】

【 0 0 0 5 】

しかし、上述の従来技術では、すべての触覚センサ情報がホストコンピュータに送られているので、触覚センサが高密度に分布されて高サンプリングレートで読み取られると、センサ情報が膨大になる。したがって、転送がボトルネックとなり、ホストコンピュータでの処理も膨大となってしまう。

30

【 0 0 0 6 】

それゆえに、この発明の主たる目的は、ホストコンピュータでの情報処理の負荷を軽減できる、皮膚センサネットワークを提供することである。

【 0 0 0 7 】

この発明の他の目的は、転送のボトルネックを回避できる、皮膚センサネットワークを提供することである。

【課題を解決するための手段】

【 0 0 0 8 】

請求項1の発明は、少なくとも1つの触覚センサをそれぞれが備えかつ少なくとも1つの他のノードと互いに接続された複数のノードと、少なくとも1つのノードと接続されたホストコンピュータとを含む皮膚センサネットワークである。ホストコンピュータは、接続されたノードに経路要求を送信する第1経路要求送信手段、経路要求に応じて送信される経路情報を接続されたノードから受信して第1記憶手段に記憶する経路情報取得手段、経路情報に基づいて複数のノードのそれぞれに対する触覚情報要求を接続されたノードに送信する触覚情報要求送信手段、および触覚情報要求に応じて送信される触覚情報を接続されたノードから受信して第1記憶手段に記憶する触覚情報取得手段を含む。複数のノードのそれぞれは、触覚センサの出力データを検出して第2記憶手段に記憶するセンサ出力検出手段、経路要求を受信したとき経路確保するか否かを判定する経路確保判定手段、経路確保判定手段によって経路確保すると判定されたとき当該経路要求を受信したポートをホスト側経路として第2記憶手段に記憶する経路記憶手段、経路確保判定手段によって経

40

50

路確保すると判定されたとき承諾応答をホスト側経路に送信する応答送信手段、承諾応答を受信したとき当該承諾応答を受信したポートに経路確保確認を送信する確認送信手段、経路確保確認を受信したときホスト側経路以外に経路要求を送信する第2経路要求送信手段、第2経路要求送信手段によって送信した経路要求に対する承諾応答がないと判定されるとき、自分の経路情報をホスト側経路に送信する第1経路情報送信手段、経路情報を受信したことに応じて、自分の経路情報に受信した経路情報を結合することによって生成した経路情報をホスト側経路に送信する第2経路情報送信手段、触覚情報要求を受信したことに応じて、センサ出力検出手段によって検出された出力データを触覚情報としてホスト側経路に送信する第1触覚情報送信手段、および触覚情報を受信したとき当該触覚情報をホスト側経路に送信する第2触覚情報送信手段を含む。

10

【0009】

請求項1の発明では、皮膚センサネットワークはホストコンピュータと複数のノードを含み、これらは相互接続型のネットワークを形成する。各ノードは少なくとも1つの触覚センサを備え、実施例では複数の触覚センサを備えることによって分布型高密度皮膚センサを構築する。各ノードはセンサ出力検出手段を備えており、検出された出力データを記憶する。触覚情報の検出を行うとき、ホストコンピュータの第1経路要求送信手段は、接続されたノードに経路要求を送信する。ノードの経路確保判定手段は、経路要求を受信したとき経路確保するか否かを判定する。経路記憶手段は、経路確保判定手段によって経路確保すると判定されたとき当該経路要求を受信したポートをホスト側経路として第2記憶手段に記憶する。たとえば、ホストコンピュータは経路要求の送信のたびに異なるネットワーク管理番号を送信する。ノードは経路確保するとき当該管理番号を記憶しておくので、受信した経路要求の管理番号が記憶していた値とは異なるとき、ノードは当該管理番号の経路要求を初めて受信したことになり、したがって、その受信ポートをホスト側経路として確保する。ノードの応答送信手段は、経路確保判定手段によって経路確保すると判定されたとき承諾応答をホスト側経路に送信する。また、ノードの確認送信手段は、承諾応答を受信したとき当該承諾応答を受信したポートに経路確保確認を送信する。このようにノードは、受信した経路要求に応じて承諾応答を返信し、その応答に対する確認をさらに受信する。したがって、ノード間が送受信可能なリンクで接続されていることを確認することができるので、当該リンクをホスト側経路として確保してよいことを把握できる。そして、ノードの第2経路要求送信手段は、経路確保確認を受信したとき、確保したホスト側経路以外に経路要求を送信する。このようにして、ホストコンピュータに端を発する経路要求が、順次ノードに送信されて、やがて全てのノードに送信されることとなる。ネットワークの末端に位置するノードは、経路要求を送信しても承諾応答を受信することができないので、承諾応答なしに基づいて自分がネットワークの末端であることを判定できる。この末端のノードからホスト側経路のノードに向けて経路情報の送信が開始される。つまり、ノードの第1経路情報送信手段は、第2経路要求送信手段によって送信した経路要求に対する承諾応答がないと判定されるとき、自分の経路情報をホスト側経路に送信する。自分の経路情報は、たとえば、確保経路としてホスト側ポート番号、ホストまでに介在するノード数情報（ホップ番号）等を含む。そして、ノードの第2経路情報送信手段は、経路情報を受信したことに応じて、自分の経路情報に受信した経路情報を結合することによって生成した経路情報をホスト側経路に送信する。これにより、末端に位置しないノードは、経路情報の受信に応じて、自分の経路情報と全ての末端側ノードからの経路情報を、経路情報としてホスト側経路に送信する。送信される経路情報は、たとえば送信しようとするノードを根とするツリー構造を示す。やがては、ホストコンピュータに接続されたノードが、経路情報をホスト側経路にすなわちホストコンピュータに送信する。ホストコンピュータの経路情報取得手段は、経路要求に応じて送信される経路情報を接続されたノードから受信して第1記憶手段に記憶する。このようにして、ホストコンピュータからの明示的な指定無しに、各ノードが協調してホスト側の経路を確保していくことによってデータ送信経路を確保することができる。つまり、この皮膚センサネットワークは自己組織化可能であり、経路要求の度に経路を確保することができ、断線しても経路を再構築する

20

30

40

50

ことができる。したがって、ホストコンピュータでの情報処理の負荷を軽減できるとともに、頑健な情報伝達経路を簡単に構築することができる。続いて、ホストコンピュータの触覚情報要求送信手段は、経路情報に基づいて複数のノードのそれぞれに対する触覚情報要求を接続されたノードに送信する。実施例では、触覚情報要求として各ノードの識別情報が送信される。ホストコンピュータは、経路情報に基づいてネットワーク全体の経路を把握することができるので、各ノード宛に触覚情報要求を送信することができる。ノードの第1触覚情報送信手段は、触覚情報要求を受信したことに応じて、触覚センサの出力データを触覚情報としてホスト側経路に送信する。ノードの第2触覚情報送信手段は、触覚情報を受信したとき当該触覚情報をホスト側経路に送信する。これによって、各ノードからの触覚情報は、確保された経路上をホストコンピュータまで送信される。そして、ホストコンピュータの触覚情報取得手段は、触覚情報要求に応じて送信される触覚情報を接続されたノードから受信して第1記憶手段に記憶する。このようにして、ホストコンピュータは確保された経路を通して各ノードからの触覚情報を取得することができる。

10

【0010】

請求項2の発明は、請求項1の発明に従属し、各ノードは、センサ出力検出手段によって検出された出力データを圧縮する圧縮手段をさらに含み、第1触覚情報送信手段は、触覚情報要求を受信したことに応じて、圧縮手段によって圧縮されたデータを触覚情報としてホスト側経路に送信する。

【0011】

請求項2の発明では、各ノードは圧縮手段を備えており、検出された出力データを圧縮する。たとえば、触覚情報の局所性を利用して空間でのデータ圧縮が行われ、また時間方向でのデータ圧縮やノイズ除去による圧縮が行われてよい。ノードの第1触覚情報送信手段は、触覚情報要求を受信したことに応じて、圧縮手段によって圧縮されたデータを触覚情報としてホスト側経路に送信する。したがって、各ノードでの圧縮によってデータ量を減少させた触覚情報を送信できるので、転送のボトルネックを回避できるし、ホストコンピュータでの情報処理の負荷を軽減できる。

20

【0012】

請求項3の発明は、請求項2の発明に従属し、各ノードは、センサ出力検出手段で検出した触覚センサの出力データを当該触覚センサの隣接センサの所属するノード宛に送信する隣接センサ出力送信手段、および隣接センサの所属するノードからの当該隣接センサの出力データを受信して第2記憶手段に記憶する隣接センサ出力取得手段をさらに含む。圧縮手段は、触覚センサの出力データをマスクするか否かを、センサ出力検出手段によって検出されたまたは隣接センサ出力取得手段によって記憶された当該触覚センサの隣接センサの出力データに基づいて判定する第1マスク判定手段を含む。第1触覚情報送信手段は、第1マスク判定手段によってマスクしないと判定された触覚センサの出力データを、触覚情報として送信する。

30

【0013】

請求項3の発明では、隣接センサ出力送信手段と隣接センサ出力取得手段によって、各隣接センサの所属するノード間で隣接センサの出力データが交換される。実施例では、各ノードは隣接センサの所属するノードに関する位置情報を記憶しており、この位置情報に基づいて隣接センサのデータを送受信する。したがって、所属ノードによる境界を挟んで存在する隣接センサの出力データを取得することができる。圧縮手段の第1マスク判定手段は、各触覚センサの出力データをマスクするか否かを、当該触覚センサの隣接センサの出力データに基づいて判定する。たとえば触覚センサの出力値が隣接センサの出力値に基づいて予測可能な場合には、当該データをマスクする、すなわち、ホストへ送信しないと判定する。第1触覚情報送信手段は、マスクしないと判定された触覚情報を送信する。したがって、触覚情報の局所性を利用して、空間方向でデータを圧縮することができ、転送データ量を低減することができる。

40

【0014】

請求項4の発明は、請求項3の発明に従属し、第1マスク判定手段は、隣接センサの出

50

力値の平均を中心とするかつ出力値の差に第1係数を掛けた値を幅とする第1マスク領域を算出するマスク領域算出手段を含み、触覚センサの出力値が第1マスク領域内にあるとき、当該触覚センサの出力データをマスクすると判定し、第1マスク領域内にないときは当該出力データをマスクしないと判定する。

【0015】

請求項4の発明では、第1マスク判定手段はマスク領域算出手段を含み、隣接センサの出力値の平均と中心とし、出力値の差に第1係数を掛けた値を幅とする第1マスク領域を算出する。そして、触覚センサの出力値が第1マスク領域内にあるとき、当該出力データをマスクすると判定し、第1マスク領域外のときはマスクしないと判定する。平均値付近にあるデータは他のデータから予測可能な冗長性を含むので、そのような内挿可能な出力データをマスクすることができる。また、第1係数の設定値を調整することによって、圧縮率や転送されない情報量を簡単に調整することができる。各ノードまたは各触覚センサで適宜に第1係数を設定することによって、たとえば、敏感な部分と鈍感な部分を皮膚センサに設けることができる。

10

【0016】

請求項5の発明は、請求項4の発明に従属し、第1マスク判定手段は、隣接センサの出力値に対する触覚センサの出力値の変化量が第1閾値よりも小さいとき、当該触覚センサの出力データをマスクすると判定する第1ノイズマスク判定手段をさらに含む。

【0017】

請求項5の発明では、第1ノイズマスク判定手段は、空間で隣接するセンサに対する変化量が第1閾値よりも小さいとき、出力データをマスクすると判定する。したがって、変化量がノイズレベルの触覚情報をマスクすることができ、ノイズによる無駄な情報転送を低減できる。また、第1閾値の設定値を調整することによって、上述の第1係数の場合と同様に、圧縮率やマスクする情報量を調整することが可能である。

20

【0018】

請求項6の発明は、請求項2ないし5の発明のいずれかに従属し、圧縮手段は、触覚センサの出力データをマスクするか否かを、センサ出力検出手段によって検出された当該触覚センサの出力データの履歴に基づいて判定する第2マスク判定手段を含む。第1触覚情報送信手段は、第2マスク判定手段によってマスクしないと判定された触覚センサの出力データを、触覚情報として送信する。

30

【0019】

請求項6の発明では、圧縮手段の第2マスク判定手段は、触覚センサの出力データをマスクするか否かを、当該触覚センサの出力データの履歴に基づいて判定する。たとえば触覚センサの出力値が当該センサの過去の出力値に基づいて外挿により予測可能な場合には、当該データをマスクする、すなわち、ホストへ送信しないと判定する。第1触覚情報送信手段は、マスクしないと判定された触覚情報を送信する。したがって、時間方向でデータを圧縮することができ、転送データ量を低減することができる。

【0020】

請求項7の発明は、請求項6の発明に従属し、第2マスク判定手段は、触覚センサの過去の出力値からの変化量が第2閾値よりも小さいとき、当該触覚センサの出力データをマスクすると判定する第2ノイズマスク判定手段をさらに含む。

40

【0021】

請求項7の発明では、第2ノイズマスク判定手段は、過去の履歴からの変化量が第2閾値よりも小さいとき、出力データをマスクすると判定する。したがって、過去からの変化がノイズレベルの触覚情報をマスクすることができ、無駄な情報転送を低減できる。また、上述の第1閾値の場合と同様に、第2閾値の設定値の調整によって、転送情報量を調整することができる。

【0022】

請求項8の発明は、請求項1ないし7の発明のいずれかに従属し、各ノードは、触覚情報を受信したとき送信データの優先順位を判定する優先順位判定手段をさらに含む。第2

50

触覚情報送信手段は、優先順位判定手段によって受信した触覚情報の優先順位が高いと判定されたとき、当該受信した触覚情報を優先的に送信する。

【 0 0 2 3 】

請求項 8 の発明では、各ノードは、触覚情報を受信したとき、優先度の高いものを優先的にホスト側経路に送信することができる。したがって、ホストコンピュータは優先度の高い触覚情報を先に取得できるので、たとえば緊急度の高い触覚情報に素早く対応することが可能になる。

【 0 0 2 4 】

請求項 9 の発明は、少なくとも 1 つの触覚センサをそれぞれが備えかつ少なくとも 1 つの他のノードと互いに接続された複数のノードと、少なくとも 1 つのノードと接続されたホストコンピュータとを含む皮膚センサネットワークである。ホストコンピュータは、接続されたノードに経路要求を送信する第 1 経路要求送信手段、および経路要求に応じて送信される触覚情報を含んだ経路情報を接続されたノードから受信して第 1 記憶手段に記憶する情報取得手段を含む。複数のノードのそれぞれは、触覚センサの出力データを検出して第 2 記憶手段に記憶するセンサ出力検出手段、センサ出力検出手段によって検出された出力データを圧縮する圧縮手段、経路要求を受信したとき経路確保するか否かを判定する経路確保判定手段、経路確保判定手段によって経路確保すると判定されるとき当該経路要求を受信したポートをホスト側経路として第 2 記憶手段に記憶する経路記憶手段、経路確保判定手段によって経路確保すると判定されるとき承諾応答をホスト側経路に送信する応答送信手段、承諾応答を受信したとき当該承諾応答を受信したポートに経路確保確認を送信する確認送信手段、経路確保確認を受信したときホスト側経路以外に経路要求を送信する第 2 経路要求送信手段、第 2 経路要求送信手段によって送信した経路要求に対する承諾応答がないと判定されるとき、圧縮手段による圧縮に基づく触覚情報を含んだ自分の経路情報をホスト側経路に送信する第 1 情報送信手段、および経路情報を受信したことに応じて、圧縮手段による圧縮に基づく触覚情報を含んだ自分の経路情報に受信した経路情報を結合することによって生成した、触覚情報を含んだ経路情報をホスト側経路に送信する第 2 情報送信手段を含む。

【 0 0 2 5 】

請求項 9 の発明では、上述の請求項 2 の発明とは、経路情報を送信する際に触覚情報を含めて送信する点が異なっている。具体的には、ノードの第 1 情報送信手段は、自分がネットワークの末端であることを判定したとき、圧縮手段による圧縮に基づく触覚情報を含んだ自分の経路情報をホスト側経路に送信する。ノードの第 2 情報送信手段は、経路情報を受信したことに応じて、圧縮手段による圧縮に基づく触覚情報を含んだ自分の経路情報に受信した経路情報を結合することによって生成した、触覚情報を含んだ経路情報をホスト側経路に送信する。これにより、末端に位置しないノードは、経路情報の受信に応じて、自分の経路情報（触覚情報を含む）と全ての末端側ノードからの経路情報（触覚情報を含む）を、触覚情報を含んだ経路情報としてホスト側経路に送信する。この送信データは、たとえば送信しようとするノードを根とするツリー構造を示す。したがって、ホストコンピュータの情報取得手段は、経路要求に応じて送信される触覚情報を含んだ経路情報を接続されたノードから受信して第 1 記憶手段に記憶する。したがって、上述の発明と同様に、自己組織化可能なセンサネットワークを形成でき、頑健な情報伝達経路を簡単に構築できる。さらに、各ノードでの圧縮によってデータ量を減少させた触覚情報を送信できるので、転送のボトルネックを回避できるし、ホストコンピュータでの情報処理の負荷を軽減できる。

【 発明の効果 】

【 0 0 2 6 】

この発明によれば、触覚センサをそれぞれが備える複数のノードとホストコンピュータとでセンサネットワークを構成し、各ノードが相互に接続された経路により通信を行うようにした。ノード間の相互接続による通信によって、自己組織化可能なセンサネットワークを形成することができる。したがって、ホストからの明示的な経路指示の必要がないの

10

20

30

40

50

でホストコンピュータでの処理負担を軽減できるとともに、頑丈な情報伝達経路を簡単に構築することができる。

【0027】

また、たとえば、隣接する触覚センサの出力をノード間で相互に通信することによって、各ノードにおいて触覚情報の局所性を利用した空間方向圧縮を行うことができるし、また、各ノードにおいて触覚情報の時間方向圧縮を行うことができる。このような各ノードでのデータ圧縮によって、転送する触覚情報のデータ量を減少させることができる。したがって、転送のボトルネックを回避することができ、ホストコンピュータでの情報処理の負担を軽減することができる。

【0028】

この発明の上述の目的、その他の目的、特徴および利点は、図面を参照して行う以下の実施例の詳細な説明から一層明らかとなる。

【発明を実施するための最良の形態】

【0029】

図1を参照して、この実施例の皮膚センサネットワーク10は、ホストコンピュータ12と複数のノード14を含む。ホストコンピュータ12および複数のノード14は相互接続型のネットワークを形成している。各ノード14は他の少なくとも1つのノード14と相互に接続されている。この実施例では各ノード14は4つの通信ポートを備えており、最大で4つのノードと接続され得る。各ノード14はこの相互に接続された経路を介して通信を行う。ホストコンピュータ12は、ネットワークを形成する複数のノード14のうち

【0030】

の少なくとも1つのノード14と接続されている。

【0031】

この皮膚センサネットワーク10は、たとえば人の腕や手（図1で破線で示される）等の物に触れられたときの触覚情報を検出するためのシステムである。

【0032】

各ノード14にはこの実施例では複数の触覚センサ16が接続されている。各ノード14は、接続されている複数の触覚センサ16のそれぞれの出力を検出し、センサ出力データとして取得する。なお、ノード14は少なくとも1つの触覚センサ16を備えていればよい。

【0033】

この皮膚センサネットワーク10が日常生活環境で人間と共存しコミュニケーションするコミュニケーションロボットに設けられる場合には、ホストコンピュータ12のプロセサ（CPU）は当該コミュニケーションロボットの全体的な制御を行うプロセサであってよい。

【0034】

図2にはノード14の電氣的構成の一例が示される。ノード14は基板18（破線表示）を含み、基板18にはこのノード14の制御を行うCPU20が設けられる。一例として、基板18の大きさは70×60mm程度であり、CPU20にはたとえばルネサス製H8S2633が使用され得る。このプロセサはCISC構成の命令を69種類実行可能であり、センサデータの処理をノード上で行える。CPU20はメモリ22と接続され、メモリ22はセンサデータバッファ24および転送データバッファ28と接続される。メモリ22はROMおよびRAM等を含み、ROMにはノード14の動作を制御するための

10

20

30

40

50

プログラムおよびデータが予め記憶されている。CPU 20はメモリ 22のROMのプログラムに従ってデータをRAMに生成または取得しつつ処理を実行する。たとえば、メモリ 22は、内蔵ROM 256 kByte、RAM 16 kByteを含み、クロック 12.288 MHzをもとに動作する。

【0035】

センサデータバッファ 24にはA/D変換器 26が接続される。A/D変換器 26としては、たとえば15チャンネル10ビットのものが使用され得る。つまり、この実施例では、1つのA/D変換器 26には最大15個の触覚センサ 16が接続され得る。A/D変換器 26は、触覚センサ 16から出力された信号をデジタルデータに変換する。A/D変換器 26は、センサ出力データを、この実施例では0を基準点として - 512から512 10までの値で出力する。そのセンサ出力データはセンサデータバッファ 24に与えられ、センサ 16ごとに設けられた領域に格納される。CPU 20はセンサデータバッファ 24からセンサ出力データを取得して、センサ出力データを、当該データを出力した触覚センサ 16の識別情報に関連付けてメモリ 22に格納する。センサ出力データはたとえばCPU 20の内部タイマに基づく検出時刻を付与されて記憶されてよい。センサ出力のサンプリングレートはたとえば100 Hzなど適宜に設定される。

【0036】

転送データバッファ 28には複数のシリアル通信のためのポート 30が接続される。この実施例では4つのポートが設けられる。各ポート 30で独立全二重通信が可能であり、通信速度はたとえば38400 bpsとした。各ノード 14はポート 30およびケーブル 20を介して他のノード 14（もしくはホストコンピュータ 12）と接続される。CPU 20は、送信すべきデータをメモリ 22から転送データバッファ 28の所定の領域に与えて、対応するポート 30から当該データを送信する。また、他のノード 14（もしくはホストコンピュータ 12）から送信されたデータは、ポート 30を介して転送データバッファ 28の所定の領域に格納される。CPU 20は転送データバッファ 28から受信データを取得して、受信データを、たとえば当該データを受信したポート 30の識別情報に関連付けてメモリ 22に格納する。

【0037】

触覚センサ 16としては、たとえば piezofilm（圧電センサ）が用いられる。piezofilmはたとえば米国MSI社製、株式会社東京センサ販売のものを使用してよい（<http://www.t-sensor.co.jp/PIEZO/TOP/index.html>）。このpiezofilmは、圧電フィルム（たとえばPVD F（ポリビニリデンフルオライド））の両面に金属薄膜が形成された構造、つまり、圧電体が導体で挟まれた構造を有する圧電センサである。piezofilmは、圧力等で変形されると、そのひずみ速度に応じた電圧を発生する。 30

【0038】

この触覚センサ 16すなわちpiezofilmが、図3に示すように、柔軟な素材からなる皮膚 32に埋め込まれることによって、触覚センサデバイスが構成される。この実施例では、皮膚 32は、下層のウレタンスポンジ 34と、その上に積層されるシリコンゴムシート 36とを含む。これらウレタンスポンジ層 34とシリコンゴムシート層 36との間にpiezofilm 16が設けられる。一例として、ウレタンスポンジ層 34の厚みは1 mm 40程度であり、シリコンゴムシート層 36の厚みも1 mm程度である。また、piezofilm 16は、たとえば、30 x 30 mm程度の大きさにカットされ、約5 mmの間隔で皮膚 32中に配置される。

【0039】

なお、皮膚 32の構成は上述のものに限定されるものではなく適宜変更され得る。たとえば本件出願人による特開2004 - 283975号に開示したように、シリコンゴムシート層を2層で形成し、piezofilmをシリコンゴムシートの比較的薄い上層と比較的厚い下層との間に配置するようにしてもよい。

【0040】

この皮膚センサネットワーク 10のような相互接続型のネットワークでは、ホストコン 50

コンピュータ12までの経路を確保する必要がある。ホストコンピュータ12が各ノード14に経路をマクロキャリブレーションによって明示的に示す場合には、ホストコンピュータ12に負担がかかる。そのため、この実施例では、各ノード14が協調して経路確保を行う。このようにホストコンピュータ12からの明示的な指定なしに各ノード14がデータ送信経路を確保できるとき、そのセンサネットワークは自己組織化可能であると表現する。この皮膚センサネットワーク10は、相互接続型の自己組織化可能なセンサネットワークに基づく分布型高密度皮膚センサを構成する。

【0041】

自己組織化可能なセンサネットワークは、ホストコンピュータからは1つのデバイスのように使うことができ、ホストコンピュータはネットワークを意識しなくてもよい。発明者等は、アドホックネットワークのDSR (Dynamic Source Routing Protocol) と類似した手法を用いて、この自己組織化ネットワークのためのプロトコルを作成した。この作成したプロトコルによってホストコンピュータ12まで転送する経路を明示的指示なしに確保できることを、発明者等はシミュレータで確認している。

10

【0042】

概略を言えば、図4(A)に示すように、まず、ホストコンピュータ12が経路確保の命令(経路要求コマンドREQ)をブロードキャスト送信する。各ノード14は、経路確保命令を受信すると、他のポート30にそのコマンドのコピーを送信する。したがって、ホストコンピュータ12からの経路確保命令は全てのノード14に送信されることとなる。

20

【0043】

各ノード14は、ブロードキャストの信号(経路要求コマンド)が流れてきた方向に経路確保を行う。たとえば、各ノード14のCPU20は、最初に経路確保命令を受信したポート30をホスト側経路として確保し、つまり、そのポート30の識別情報(ポート番号)を、ホスト側の経路を示す経路確保データとしてメモリ22に記憶する。ノード14のCPU20は、ホストコンピュータ12宛にデータを転送する際には、メモリ22に記憶した経路確保データを参照して、ホスト側のポート30に向けてデータを送信する。このようにして、たとえば図4(B)に示すように、皮膚センサネットワーク10では、ホストコンピュータ12からの明示的な経路の指定なしに、通信の経路を確保することができる。このように、皮膚センサネットワーク10は、相互接続型の自己組織化センサネットワークを構築する。

30

【0044】

また、ネットワーク中でたとえば断線があった場合でも、上述のようにホストコンピュータ12から経路確保の命令を送信することによって各ノード14が経路確保を行うので、図4(C)に示すように、通信経路を簡単に再構築することができる。

【0045】

この手法により、ホストコンピュータ12はネットワークの構造を意識することなく、キャリブレーションの信号(経路確保要求)を送ってデータを受信するという2つのステップでセンサネットワークを使うことができる。

【0046】

具体的には、各ノード14は、上述のように、独立に通信できる複数の通信ポート30を有し、受信データについて識別可能である。経路確保とデータ転送のために、次の(1)から(8)に示すようなコマンドを設ける。各ノード14は、コマンド受信時には当該コマンドを解釈し、決められたルールに従って動作する。

40

【0047】

(1)経路要求コマンドREQは、経路確保を要求するためのコマンドである。経路要求コマンドは初めにホストコンピュータ12から送信される。また、各ノード14は、ホスト側のノード14と双方向通信が確認されたことに応じて、この経路要求コマンドをホスト側以外のポート30に向けて送信する。経路要求コマンドは、引数として世代番号Gnおよびホップ番号Hnを含む。世代番号Gnおよびホップ番号Hnはメモリに記憶さ

50

れている。

【 0 0 4 8 】

なお、世代番号 G_n はネットワーク世代番号であり、ルーティングのためのネットワーク管理番号であり、初期値は 0 である。ホストコンピュータ 1 2 から経路確保要求するたびにその値が変更される。各コマンドは引数としてこの世代番号 G_n を必ず含んで送信され、各ノード 1 4 ではこの世代番号 G_n が常にチェックされる。経路要求コマンドの場合には、この世代番号 G_n が異なるとき経路が再確保されることとなり、他のコマンドの場合にはこの世代番号 G_n はネットワークの識別に用いられる。

【 0 0 4 9 】

また、ホップ番号 H_n は、ホストコンピュータ 1 2 からのホップ数を示し、初期値は 0 である。ホップ番号 H_n は、今の経路がこの番号分のノード 1 4 を介してホストに経路を確保していることを示す。ホストコンピュータ 1 2 のポートと直接接続されているノード 1 4 のホップ番号 H_n は 1 になる。ホップ番号 H_n はたとえば経路データの結合時に使用される。

【 0 0 5 0 】

この経路要求コマンドを受信したとき、ノード 1 4 は世代番号 G_n を確認する。その世代番号 G_n の含まれるコマンドをまだ受信していない場合には、ノード 1 4 は当該経路要求コマンド受け取ったポート 3 0 に向けて経路応答コマンド `REP__local` (承諾) を送信する。これによって、経路要求を承諾し、経路応答確認コマンドを待つ。一方、その世代番号 G_n の含まれるコマンドを既に受信している場合には、ノード 1 4 は当該経路要求コマンドを受け取ったポート 3 0 に向けて経路応答コマンド `REP__local` (拒否) を送信して、経路要求を拒否する。

【 0 0 5 1 】

(2) 経路応答コマンド `REP__local` は、ルート要求に対する承諾または拒否の応答をするためのコマンドである。経路応答コマンドは引数として世代番号 G_n およびホップ番号 H_n を含む。経路応答コマンド (承諾) を受信したとき、ノード 1 4 (またはホストコンピュータ 1 2) は、経路確保確認コマンドを受信ポート 3 0 に向けて送信する。そして、ネットワーク経路応答コマンドを待つ。一方、経路応答コマンド (拒否) を受信したとき、ノード 1 4 は当該ポート 3 0 に向けた経路要求に対する拒否を記憶する。

【 0 0 5 2 】

(3) 経路応答確認コマンド `CONFIRM__REP` は、ルート確立の確認をするためのコマンドである。経路応答確認コマンドは引数として世代番号 G_n およびホップ番号 H_n を含む。経路応答確認コマンドを受信したとき、ノード 1 4 は経路要求コマンドを受信したポート 3 0 以外の全てのポート 3 0 に向けて経路要求コマンドを送信する。そして、経路応答コマンドを待つ。

【 0 0 5 3 】

なお、経路応答コマンド待ち処理で、タイムアウトになったとき、または全てのポート 3 0 から経路応答コマンド (拒否) を受信したときには、当該ノード 1 4 がネットワークの末端になる。したがって、当該末端のノード 1 4 は、自分の経路データを生成して、ホスト側のポート 3 0 に向けてネットワーク経路応答コマンドを送信する。

【 0 0 5 4 】

(4) ネットワーク経路応答コマンド `REP__entire` は、ルート情報をホストに知らせるためのコマンドである。ネットワーク経路応答コマンドは引数としてデータ長 `Dlen`、世代番号 G_n およびデータ `Data` を含む。ネットワーク経路応答コマンドを受信したとき、ノード 1 4 は自分の経路情報を付加した経路データを生成する。末端側 (葉側) の全てのポート 3 0 からネットワーク経路応答コマンドを受信した場合には、ノードは生成した経路データを含むネットワーク経路応答コマンドをホスト側のポート 3 0 へ向けて送信する。

【 0 0 5 5 】

(5) ホスト中継コマンド `RELAY__to__Host` は、センサデータをホストコン

10

20

30

40

50

コンピュータ12に送るためのコマンドである。ホスト中継コマンドは引数としてデータ長 `Dlen`、世代番号 `Gn` およびデータ `Data` を含む。ホスト中継コマンドを受信したとき、ノード14は、確保した経路（ホスト側経路）のポート30に向けて、当該受信したデータのコピーを送信する。

【0056】

なお、この実施例のように、ホスト中継コマンドを受信したときには、データの優先順位を判定し、優先順位の高いデータを先に送信するようにしてもよい。この場合には、たとえば緊急に送信したいデータを先にホストコンピュータ12に送信することができる。

【0057】

(6) ノード中継コマンド `RELAY__to__Node` は、任意のノードにコマンドまたはデータを送信するためのコマンドである。ノード中継コマンドは引数としてデータ長 `Dlen`、世代番号 `Gn`、指定ポート番号 `Pn` およびデータ `Data` を含む。このノード中継コマンドを受信したときには、受信したデータがコマンドであれば、ノード14は、指定された番号のポートに向けて当該コマンドを送信する。一方、受信したデータがコマンドでなければ、当該データを指定番号のポート30に向けてノード中継コマンドで送信する。また、指定ポート番号 `Pn` が `localhost` のストリームへのポート番号である場合には、ノード14は自分自身に対して送信されたデータであると判断して、当該データを受信して記憶する。

【0058】

このノード中継コマンドは、ホストと任意ノード間通信のため、および、任意ノード間（隣接ノードを含む）通信のために設けられている。ポート番号 `Pn` をたとえば1, 2, ..., `n` で割り振り、`localhost` のストリームへのポート番号をたとえば0とする。たとえばノード1（ポート__2）ノード2という経路（ノード1はポート__2を介してノード2と接続されている）の場合において、ノード1が `RELAY__to__Node` (`Dlen`, `Gn`, `Pn`=0, `Data`=(センサ出力データ)) をポート__2へ送信したとき、ノード2は、指定ポート番号 `Pn` が0なので、受信した「`Data` (センサ出力データ)」が自分自身へ送られたものであると判断できる。したがって、隣接ノード14に対してセンサ出力データを送信することができる。また、たとえばノード1（ポート__2）ノード2（ポート__1）ノード3という経路の場合において、ノード1が `RELAY__to__Node` (`Dlen`, `Gn`, `Pn`=1, `Data`=(`"Pn=0"`+(センサ出力データ))) をポート__2に向けて送信すれば、ノード1は自身のセンサ出力データをノード3に対して送信することができる。

【0059】

なお、上述では、ノード14が自己に所属する触覚センサ16のセンサ出力データを検出したとき、隣接センサ位置情報に基づいて相手のノード14（隣接センサの所属するノード14）にセンサ出力データを送信する場合を説明した。しかし、相手のノード14からのリクエストに応じてセンサ出力データを送信するようにしてもよい。このように隣接ノードを超えて物理的に近い位置に存在するセンサ16のセンサデータを交換するためには、ネットワークの経路情報は必要ではないが、当該センサの位置情報を記憶しておく必要がある。また、このコマンドを使って任意のノード間でデータを与える場合には、ホストが経路を指示する必要がある。

【0060】

(7) 識別番号割当コマンド `GIVE__ID__NUMBER` は、識別番号を各ノード14に割り振るためのコマンドである。識別番号割当コマンドは引数として世代番号 `Gn`、ホップ番号 `hn` およびデータ `Data` を含む。この識別番号割当コマンドを受信したとき、ノード14はデータに含まれる識別番号を記憶する。識別番号割当コマンドは、ホストコンピュータ12から各ノード14に向けて送信される。ホストコンピュータ12は経路データを受信したとき、経路データに基づいて経路上の各ノード14の識別番号を特定する。ホストコンピュータ12のメモリには、たとえばノード配置情報が予め記憶されていて、各ノード14の識別番号と各ノード14が他のノード14との接続に使用しているポ

10

20

30

40

50

ート番号等が記憶されている。ホストコンピュータ12はこのノード配置情報と取得した経路情報に基づいて、取得された経路上における各ノード14の識別番号を特定することができる。そして、ホストコンピュータ12は、自己に直接接続されたノード14にはこの識別番号割当コマンドを用いて当該ノード14の識別番号を送信し、他のノード14には、ノード中継コマンドを用いてかつそのデータにこの識別番号割当コマンドを含めることによって、各ノード14に当該ノード14の識別番号を送信する。

【0061】

たとえば、ノード1のポート2に接続されているノード2に対して識別番号を与える際には、ホストコンピュータ12は、`RELAY__to__Node(Dlen, Gn, Pn = 2, Data = (GIVE__ID__NUMBER(Hn, Gn, Data = (識別番号)))`をノード1の接続されているポートに向けて送信する。これに応じてノード1は、`Data`がコマンドなので、指定された番号`Pn (= 2)`のポートへ向けて、当該識別番号割当コマンドを送信する。したがって、ノード2は識別番号割当コマンドを受信して、識別番号を記憶することができる。

10

【0062】

また、この識別番号割当コマンドを受信したことに応じて、ノード14はセンサデータ(触覚情報)をホストへ送信するための処理を開始する。すなわち、この識別番号割当コマンドはホストコンピュータ12からのセンサデータ送信要求に相当する。

【0063】

(8)エラーコマンド`Error`は、エラーをホストに知らせるためのコマンドである。エラーコマンドは引数として世代番号`Gn`およびエラー番号`En`を含み、場合によってはデータ`Data`を含んでよい。エラーコマンドを受信したとき、ノード14はエラー番号を記憶し、当該エラー番号を既に送信したことがあるか否かを判定する。当該エラー番号をまだ受信したことがなかった場合には、ノード14は他のポート30に向けてエラーコマンドを送信し、一方、当該エラー番号を既に受信していた場合には当該エラーコマンドの送信は行わない。

20

【0064】

また、この実施例では、触覚センサ16を各ノード14が管理しているため、センサ出力情報を取得できる範囲が限定され、触覚センサ16間に境界が形成されることとなる。しかし、隣接する触覚センサ16の接続されたノード14と相互通信して情報を取得することによって、各ノード14がその境界を越えてセンサ出力情報を取得することが可能になる。相互接続によるセンサネットワークはお互いのノード14間の通信に有利であるため、高い時間分解能で隣接するノード14から情報を取得することができ、しかも、このノード14間相互通信がネットワーク全体の通信を妨げるようなことはない。したがって、各ノード14は、1つの触覚センサ16の情報を時間方向で圧縮するだけでなく、複数のセンサ16の反応値に基づいて空間方向で情報を圧縮して、触覚情報を抽出することができる。

30

【0065】

触覚情報の特徴を視覚情報と比較した場合、視覚は、全体的な状況を検出するものであり、また非接触である。これに対して、触覚は物理的な接触を局所的に検出するという特徴がある。つまり、触覚は局所的な情報の集合であり、したがって、この実施例では、この局所性を利用することで触覚情報を効率的に圧縮するようにしている。

40

【0066】

図5には、空間における局所性を利用して、触覚情報の圧縮を行う方法の概要が示される。この方法では、たとえば、圧縮処理しようとする i 番目のセンサに対して、空間でのデータの内挿が可能かどうか注目し、データをマスクする(すなわちデータをホストコンピュータ12に転送しない)値の範囲を適宜に設定する。この実施例では、マスクする値の範囲は、隣接するセンサ16の反応値に基づいて設定される。マスク範囲ないし領域は、一例として、隣接センサ16のセンサ値の平均値と差によって算出される。具体的には、平均値 S_{im} を中心として、かつ、隣接センサ値の差 S_i に係数 k を掛けることに

50

よって算出された値 $k \cdot S_i$ を幅とした範囲を、データのマスク領域として設定する。

【0067】

図5の例では、 $i - 1$ 番目のセンサエレメントと $i + 1$ 番目のセンサエレメントの2つが i 番目のセンサエレメントの隣接センサである場合が示されている。 $i - 1$ 番目のセンサ値を S_{i-1} とし、 $i + 1$ 番目のセンサ値を S_{i+1} とすると、差は $S_i = \{ (S_{i+1}) - (S_{i-1}) \} / 2$ に従って算出され、平均値は $S_{im} = \{ (S_{i+1}) + (S_{i-1}) \} / 2$ に従って算出される。そして、処理しようとする i 番目のセンサ値が上記マスク範囲に含まれる場合には、当該 i 番目のセンサを管理するノード14は、その i 番目のセンサのデータをホストコンピュータ12宛に転送しない、すなわちマスクする。

【0068】

なお、マスク領域パラメータ k の値は適宜に設定される。平均値付近にあるデータは他のデータから予測可能な冗長性を含むと考えられるので、たとえば係数 k が大きいということはそのセンサの情報の冗長性を大きく見積もるということの意味する。

【0069】

また、センサの隣接は1次元に限られるものではなく、2次元、および3次元の隣接であってもよい。さらにまた、マスク領域の値は、上述の隣接センサ値の差分やパラメータ k を使用する算出方法によって設定することに限られるものではなく、触られ方などの触覚情報の把握に支障のない適切なものであれば他の方法で設定されてよい。

【0070】

また、上述のような空間方向の圧縮だけでなく、場合によっては、図6に示すように、1つのセンサエレメントのデータを時間軸方向で圧縮するようにしてよい。この圧縮は、これまでに取得したセンサ値の時間外挿に基づいて行われる。つまり、前回時刻 t でのセンサ値 S_t と前々回の検出時刻 $t - 1$ でのセンサ値 S_{t-1} に基づいて、たとえば1次線形外挿によって現時刻 $t + 1$ での値が予測可能か否かに注目し、データをマスクする範囲を設定する。マスク範囲の算出の一例として、まず、センサ値 S_t とセンサ値 S_{t-1} との差 S_t に係数 k_t を掛けることによって幅 $k_t \cdot S_t$ を算出する。なお、差は $S_t = \{ (S_t) - (S_{t-1}) \} / 2$ に従って算出される。次に、時刻 t におけるセンサ値 S_t を中心として $k_t \cdot S_t$ を幅とした範囲を設定する。そして、時刻 $t - 1$ でのセンサ値 S_{t-1} と設定範囲の下限値 $(S_{t-1} - k_t \cdot S_t)$ とに基づく外挿によって時刻 $t + 1$ における下限予測値を算出する。また、時刻 $t - 1$ でのセンサ値 S_{t-1} と設定範囲の上限値 $(S_{t-1} + k_t \cdot S_t)$ とに基づく外挿によって時刻 $t + 1$ における上限予測値を算出する。そして、算出された時刻 $t + 1$ における下限予測値から上限予測値までの範囲を、時刻 $t + 1$ におけるマスク領域として設定する。

【0071】

したがって、今回の時刻 $t + 1$ で検出したセンサ値 S_{t+1} が、設定されたマスク領域に含まれる場合には、そのセンサエレメントを管理するノード14は、当該センサ値 S_{t+1} をホストコンピュータ12宛に転送しない。この時間軸方向の圧縮によっても転送データ量を減少させることができるが、ノイズの低減に効果的である。

【0072】

また、アナログのセンサデバイスを読み取りデジタル化する際には、量子化誤差だけでなく様々な要因によってA/D変換の結果にノイズなどの影響が誤差として出てくる。そのため、 S/N 比の小さいノイズレベルの変化を観測したときには、実際の触覚デバイスからのものであるのかあるいはノイズであるのかの判断は困難で、含まれる情報量も少ない。たとえば、図7に示すように、マスク領域と隣接センサエレメントの差 S_i との関係は、傾き k の直線となる。そこにノイズレベル N を設定する。すなわち、空間で隣接するセンサ16に対して、変化量が閾値 N 以下の値となった場合には、ノイズとみなして、当該センサデータを無条件にマスクする。また、時間方向で過去の履歴から変化量がノイズレベル(閾値) N_t 以下のセンサデータも、ノイズとみなしてマスクする。これによって、ノイズによって生じる無駄な情報転送を低減できる。

【0073】

10

20

30

40

50

このようにして圧縮されてホストコンピュータ12に転送されるセンサデータは、全体としては触覚出力の変化領域の輪郭を示している。つまり、ホストコンピュータ12は、センサデータとして圧力値が転送される場合には、圧力の変化している部分の輪郭を把握することができる。

【0074】

上述のような圧縮方法では、圧縮率とマスクされる（ホストコンピュータ12に転送されない）情報量とは、マスク領域パラメータ k （ k_t ）の値と想定するノイズレベル N （ N_t ）の値で調整することができる。情報量をコントロールするにはこの2つのパラメータを適宜変更すればよい。たとえば、この皮膚センサネットワーク10をコミュニケーションロボットに設ける場合には、ロボットで敏感にしたい部分では情報量が多くなるように各パラメータを設定し、鈍感でよい部分では情報量が少なくなるように各パラメータを設定する。このようなパラメータの設定によって、たとえばロボットの触行動認識におけるアテンションコントロールを比較的容易に行うことができる。

【0075】

図8には、各ノード14のメモリ22のデータ記憶領域のメモリマップの一例が示される。ただし、図8に示されるデータは一部であり、メモリ22のデータ記憶領域には、処理に必要な各種のデータが記憶されている。

【0076】

世代番号記憶領域42には世代番号 G_n が記憶される。初期値として0が記憶される。この記憶領域42の世代番号 G_n は、ホスト側から経路要求コマンドREQを受信したときに参照される。経路要求コマンドで受信した世代番号と異なる番号が記憶されているときには、世代番号記憶領域42には受信した世代番号 G_n が記憶される。

【0077】

ホップ番号記憶領域44にはホップ番号 H_n が記憶される。初期値として0が記憶される。ホップ番号 H_n は経路確保確認コマンドの受信に応じて更新される。また、ホップ番号記憶領域44は経路データを生成するときに参照される。

【0078】

経路確保情報記憶領域46には、ホスト側のポート番号が記憶される。各ノード14がホスト側ポート番号のみを記憶することによって、ネットワーク全体で情報伝達経路が記憶されることとなる。

【0079】

識別番号記憶領域48にはホストコンピュータ12から送られてきた識別番号が記憶される。エラー番号履歴記憶領域50には、受信した（送信した）エラー番号 E_n が記憶される。エラー番号 E_n としてはエラーの種類や発生状況等に応じて異なる番号が設定されている。

【0080】

隣接センサ位置情報記憶領域52には、当該ノード14に所属する触覚センサ16と隣接する触覚センサ16の位置を示す情報が記憶される。隣接センサ位置情報は予め記憶されている。たとえば、図9に示すように、センサID（識別情報）に対応付けて、隣接センサの所属するノードに関する情報、および隣接センサの所属ノードにおけるセンサIDが記憶される。隣接センサの所属するノードに関する情報は、たとえば当該隣接センサの所属するノード14の接続されるポート番号が記憶される。隣接センサが自己のノード14に所属している場合には、所属ノードに関する情報としては、所属ノードが同じであることを示すデータが記憶される。なお、隣接センサは基本的に物理的に近い位置に存在するので、自ノードまたは隣接するノードに所属する。しかし、隣接センサが隣接ノードに所属していない場合には、当該隣接センサの所属するノードまでに、隣接ノードもしくは介在ノードのどのポートを介して接続されているかが分かるように、当該隣接センサの所属するノードまでに介在するノードおよびポート番号に関する情報を、隣接センサの所属ノードに関する情報として予め記憶しておく。また、複数の隣接センサが存在する場合には、複数の隣接センサのそれぞれの位置情報が記憶される。

【 0 0 8 1 】

センサ出力履歴記憶領域 5 4 には、自ノード 1 4 に所属する触覚センサ 1 6 の出力データの履歴が記憶される。センサ I D に対応付けられて所定の検出回数分のセンサ出力データが記憶される。なお、センサ出力データにはさらにその検出時刻を付与して記憶するようにしてよい。

【 0 0 8 2 】

隣接センサ出力記憶領域 5 6 には、他のノード 1 4 から受信した隣接センサのセンサ出力データが記憶される。たとえば、隣接センサのセンサ位置情報（所属およびセンサ I D）に対応付けて、受信したセンサ出力データが記憶される。

【 0 0 8 3 】

優先度情報記憶領域 5 8 には、データ転送の優先順位に関する情報が記憶される。優先度情報は予め記憶される。具体的には、優先的に転送すべきセンサデータを示す情報が記憶される。たとえば、痛い感覚を示すデータ、危ないことを示すデータ等のように、緊急を要するセンサデータが優先的に送信すべきデータとして設定される。また、たとえば撫でられていることを示すデータのように、所定の触られ方を示すデータを高優先度データとして設定してもよい。このようなセンサデータに対するラベリングは、センサの圧力値や設置場所等に基づいてなされる。たとえば、痛い感覚を示すデータは、センサ出力データがコミュニケーションロボットの通常のコミュニケーションではあり得ないような高いまたは異常な圧力値を示すデータであった場合にラベリングされてよい。また、危ないことを示すデータは、コミュニケーションロボットの関節の近くに配置されたセンサからの出力データに対してラベリングされてよい。この場合、各ノード 1 4 は自己のセンサ 1 6 がたとえばコミュニケーションロボットにおける関節部の近くのような危ない場所に設置されていることを示すデータをセンサ I D に対応付けて予め記憶する。

【 0 0 8 4 】

また、優先度情報は、状況に応じて変更されてもよい。この場合、ホストコンピュータ 1 2 が各ノード 1 4 に対して、状況に応じた優先度情報を、ノード中継コマンド R E L A Y _ t o _ N o d e を用いて送信する。ホストコンピュータ 1 2 がコミュニケーションロボットの C P U である場合、ホストコンピュータ 1 2 は、たとえばコミュニケーションロボットが今実行している行動や各種センサで観測される状況等に応じて、危険な場所にあるノード 1 4 およびセンサ I D をテーブルデータなどに基づいて特定し、該当するノード 1 4 に対して、該当するセンサ I D が今危ない場所になっていることを示すデータを送信してもよい。また、ホストコンピュータ 1 2 は、コミュニケーションロボットが今実行している行動や観測される状況等に応じて、優先されるべき所定の触られ方を特定し、当該所定の触られ方を示すデータを優先度情報として各ノード 1 4 に送信するようにしてもよい。

【 0 0 8 5 】

ノイズレベル情報記憶領域 6 0 には、センサデータをマスクするためのノイズレベルに相当するセンサ出力値の変化量を示す情報が記憶される。具体的には、空間方向のマスクのためのノイズレベル N と、時間方向のマスクのためのノイズレベル N_t とが記憶される。

【 0 0 8 6 】

図 1 0 には生成される経路データの内容の一例が示される。図 1 0 (A) には、ネットワークの末端のノード 1 4 がホスト側へネットワーク経路応答を送信するとき、または、各ノード 1 4 がネットワーク経路応答待ち処理を開始したときに生成される自分の経路データの内容が示されている。自分の経路データは、ホップ番号 H_n および経路として確保したホスト側のポート番号 $P_n_h o s t s i d e$ 等を含む。ホップ番号 H_n はメモリ 2 2 のホップ番号記憶領域 4 4 に記憶され、ホスト側ポート番号はメモリ 2 2 の経路確保情報記憶領域 4 6 に記憶されている。自分の経路データからは、当該ノード 1 4 が幾つのノード 1 4 を介してホストに経路を確保しているか、どのポート番号でホスト側経路につながっているかが把握される。なお、図 1 0 (A) で $s e p a r a t o r 2$ はデータの区切

10

20

30

40

50

り目を示す文字列であり、separator1はデータの終わりを示す文字列である。また、anythingは任意のデータである。つまり、ネットワーク経路応答コマンドを用いて、経路データとともに任意のデータを送信することが可能になっている。

【0087】

図10(B)には、ネットワーク経路応答コマンドを末端側(葉側)から受信したときに生成される経路データの内容の一例が示される。この経路データは、図10(A)に示した自分の経路データと受信した経路データとを結合することによって生成される。つまり、各ノード14は、ネットワーク経路応答コマンドを葉側から受信したとき、この自分の経路データに、ホップ数に関する情報TAB*Hn、および当該コマンドを受信したポート番号Pn__leafsideとともに、受信した経路データData__leafsideを追加することによって、経路データを生成する。葉側ポート番号が含まれるので、葉側の経路データをどのポートから受信したかを把握できる。この追加は、葉側のポートからのネットワーク経路応答コマンドの受信ごとに行われる。

10

【0088】

このようにして生成される経路データは、当該データを生成するノードを根とするツリー構造となる。つまり、経路データは、当該ノード14を根とした葉側のネットワークの経路を示すネットワーク経路データともいえる。各ノード14は、全ての葉側ポート30からネットワーク経路応答コマンドを受信したとき、生成した経路データをホスト側のポート30へ向けて送信する。したがって、ホストコンピュータ12は、接続されているノード14を根とするツリー構造でネットワーク全体の経路が示されたネットワーク経路データを受信することとなり、受信したネットワーク経路データによってネットワーク全体の経路を把握することができる。

20

【0089】

図11には、ホストコンピュータ12の動作の一例が示される。ホストコンピュータ12のCPUは、ステップS1で、センサデータを取得するか否かを判定する。たとえば、一定時間ごとにセンサデータを検出するような場合には検出時刻になったか否かをCPUの内部タイマに基づいて判断する。また、ホストコンピュータ12がコミュニケーションポートのCPUであるような場合には、実行する行動や観測される状況等に応じたセンサデータの検出タイミングであるか否かを判断してもよい。

30

【0090】

ステップS1で“YES”であれば、ステップS3で、世代番号Gnの値を設定する。世代番号Gnはネットワークの管理番号であり、その値は経路確保ごとに変えられ、異なる値に設定される。そして、ステップS5で、経路要求コマンドREQを、ノード14の接続されているポートへ向けて送信する。なお、経路要求コマンドで引数として世代番号Gnとともに送信されるホップ番号Hnには初期値0が設定される。

【0091】

続いて、ステップS7で、経路応答コマンドREP__local(承諾)をノード14から受信したか否かを判断する。つまり、ステップS5で送信した経路要求コマンドに応じてノード14から送信される経路応答コマンドの受信を待機する。なお、図示は省略されるが、経路応答コマンド(承諾)を受信できずにタイムアウトになった場合には、接続されているノード14との間の通信リンクに問題があると考えられるので、たとえば当該タイムアウトをメモリに記憶してステップS1に戻る。

40

【0092】

ステップS7で“YES”であれば、ステップS9で、経路確保確認コマンドCONFIRM__REPをノード14の接続されているポートへ向けて送信する。そして、ステップS11で、ステップS9で送信したコマンドに応じてノード14から送信されるネットワーク経路応答コマンドREP__entireを受信したか否かを判断する。ステップS11で“NO”であれば再びステップS11を繰り返す。つまり、ネットワーク経路応答コマンドの受信を待機する。なお、図示は省略されるが、ネットワーク経路応答コマンドを受信できずにタイムアウトになった場合には、当該タイムアウトをメモリに記憶し

50

てステップS 1に戻る。

【0093】

ステップS 11で“YES”であれば、ステップS 13で、ネットワーク経路応答コマンドに含めて送信された経路情報(ネットワーク経路情報)を抽出してメモリに記憶する。この経路情報は、ホストコンピュータ12に接続されるノード14のデータを根とするツリー構造となっている。そして、ステップS 15で、取得した経路情報に基づいて、経路上の各ノード14の識別番号を特定する。ホストコンピュータ12のメモリには、たとえば、各ノード14の識別番号、および各ノード14が他のノード14との接続に使用しているポート番号等に関する情報を含むノード配置情報が予め記憶されている。ホストコンピュータ12はこのノード配置情報と取得した経路情報に基づいて、取得された経路上における各ノード14の識別番号を特定する。

10

【0094】

そして、ステップS 17で、各ノード14にそれぞれの識別番号を送信する。この識別番号の各ノード14への送信には、上述のように、ノード中継コマンドRELAY__to__Nodeおよび識別番号割当コマンドGIVE__ID__NUMBERを用いる。経路情報を取得したので、各ノード14にその識別番号を送信することができる。この識別番号の送信は、各ノード14に対する触覚情報送信要求に相当する。

【0095】

続いて、ステップS 19では、ホスト中継コマンドRELAY__to__Hostを受信したか否かを判断する。ステップS 19で“NO”であれば再びステップS 19を繰り返す。つまり、ステップS 17の識別番号の送信に応じて各ノード14から送信されるホスト中継コマンドの受信を待機する。ステップS 19で“YES”であれば、当該コマンドに含まれるセンサデータ(触覚情報)を抽出して、メモリに記憶する。センサデータは、たとえば当該コマンドと一緒に送信されてくるノード14の識別番号および検出時刻に対応付けて記憶する。なお、検出時刻はホストコンピュータ12で付与されてもよい。また、図示は省略されるが、このホスト中継コマンドの受信待ちおよびセンサデータの記憶は、たとえばステップS 17の処理から所定時間が経過するまで継続される。この所定時間は、たとえば全てのノード14からセンサデータが送信された場合でも全センサデータを受信して記憶することが可能な値に設定されている。ステップS 21を終了するとステップS 1に戻る。

20

30

【0096】

なお、図11では省略されるが、ホストコンピュータ12のCPUは、エラーコマンドErrorを受信したときは、当該コマンドに含まれるエラー番号Enをたとえば世代番号Gnとともにメモリに記憶してステップS 1へ戻る。

【0097】

図12には各ノード14の動作の一例が示される。ノード14のCPU20は、ステップS 41で、センサ出力を取得する時刻であるか否かを判断する。たとえば所定時間間隔でセンサ出力を検出する場合には、CPU20の内部タイマに基づいて一定時間が経過したか否かが判断される。ステップS 41で“YES”であれば、ステップS 43で、センサデータバッファ24から各センサ16のセンサ出力データを取得して、センサIDおよび検出時刻等のデータに関連付けてメモリ22のセンサ出力履歴記憶領域54に記憶する。

40

【0098】

続いて、ステップS 45で、ノード中継コマンドRELAY__to__Nodeを用いて、取得したセンサ出力データを隣接センサの所属するノード14に送信する。隣接センサの所属するノード14までに経由するポート番号等の情報は、隣接センサ位置情報記憶領域52に記憶されているので、上述のようにしてノード中継コマンドでセンサ出力データを隣接センサのノード14まで送信できる。また、センサ出力データを受信したときも、ノード14は、隣接センサ位置情報とセンサ出力データに付与されたセンサIDに基づいて、隣接センサのセンサ出力データを特定できる。ステップS 45を終了し、またはステ

50

ップS 4 1で“NO”であれば、ステップS 4 7へ進む。

【0099】

ステップS 4 7では、転送データバッファ28の受信バッファをチェックし、ステップS 4 9でコマンドが記憶されているか否かを判断する。ステップS 4 9で“YES”であれば、ステップS 5 1でコマンド処理を開始する。このコマンド処理の動作は後述する図13から図28に示される。ステップS 5 1でコマンド処理を開始し、またはステップS 4 9で“NO”であれば、ステップS 5 3で全ポート30の受信バッファについて処理を済ませたか否かを判断する。ステップS 5 3で“NO”であれば、ステップS 4 7に戻って別のポート30に対する処理を実行する。ステップS 5 3で“YES”であれば、ステップS 4 1に戻る。このように、常時各ポート30の受信バッファがチェックされ、コマンドを受信しているときにはポート30ごとのコマンド処理が実行される。図12のステップS 4 1からステップS 4 5のセンサ出力の検出および隣接センサ出力データの送信、ならびに各コマンド処理は並列的に実行される。

10

【0100】

図13にはコマンド処理の動作の一例が示される。ノード14のCPU20は、ステップS 7 1で転送データバッファ28の受信バッファからコマンドをメモリ22のワークエリアに取得する。次に、ステップS 7 3で、受信コマンドが経路要求コマンドREQか否かを判断する。ステップS 7 3で“YES”であれば、ステップS 7 5で、コマンドから世代番号Gnおよびホップ番号Hnを取得ないし抽出する。

【0101】

続いて、ステップS 7 7で、取得した世代番号Gnは、メモリ22の世代番号記憶領域42に記憶された世代番号Gnと同じであるか否かを判断する。ステップS 7 7で“NO”であれば、つまり、今回設定された世代番号Gnを有する経路要求コマンドを初めて受信した場合には、ステップS 7 9で、経路確保情報記憶領域46に当該コマンドを受信したポート番号を記憶する。つまり、ホスト側の経路を示すポート番号が記憶される。また、ステップS 8 1で世代番号記憶領域42に受信した世代番号Gnを記憶する。

20

【0102】

さらに、ステップS 8 3で、受信ポートへ向けて経路応答コマンドREP_local(承諾)を送信する。そして、ステップS 8 5で、経路確保確認コマンドCONFIRM_REP待ち処理を開始する。この待ち処理は図14に示され、コマンド処理等と並列的に実行される。

30

【0103】

経路確保確認コマンド待ち処理を開始すると、図14のステップS 10 1で、経路確保確認コマンド待ちタイマのカウントをスタートする。次に、ステップS 10 3で、タイムアウトになったか否か、たとえばタイマが所定値になったか否かを判断し、“YES”であれば、ステップS 10 5で、ステータスを初期化する。具体的には世代番号記憶領域42および経路確保情報記憶領域46をクリアする。経路確保確認コマンドを受信できれば、隣接ノード14と送受信可能なリンクで繋がっているといえる。しかし、経路要求に回答したのに確認メッセージが来ないということは、送信ポートが使えず、片方向通信のみ可能なリンクであるので、他の経路を利用するべく、ステータスを初期化する。また、ステップS 10 7で、全てのポート30に向けてエラーコマンドErrorを送信する。このエラーコマンドでは、たとえば経路確保確認に失敗したことを示すエラー番号Enが送信される。なお、図示は省略されるが、送信するエラー番号はエラー番号履歴記憶領域50に記憶する。ステップS 10 7を終了すると、この経路確保確認待ち処理を終了する。

40

【0104】

一方、ステップS 10 3で“NO”であれば、ステップS 10 9で、経路確保確認コマンドを受信されたか否かを判断する。たとえば、受信したコマンドが経路確保確認コマンドであったと判定されたとき、当該コマンドの当該ポートでの受信を示すフラグをオンに設定しておく。ステップS 10 9では、この経路確保確認コマンド受信フラグに基づいて判定する。他の待ち処理でも同様に、各コマンドの各ポートにおける受信を示すフラグに

50

基づいて判定が行われてよい。ステップS 109で“NO”であれば、ステップS 103へ戻り、この待ち処理を継続する。一方、ステップS 109で“YES”であれば、この待ち処理を終了する。

【0105】

図13に戻って、ステップS 77で“YES”であれば、つまり、今回設定された世代番号Gnを有する経路要求コマンドを既に受信していた場合には、ステップS 87で、当該コマンドを受信したポート30に向けて、経路応答コマンドRREP_local(拒否)を送信する。ステップS 87を終了し、またはステップS 85で待ち処理を開始すると、このコマンド処理を終了する。なお、ステップS 73で“NO”であれば、処理は図15のステップS 121へ進む。

10

【0106】

図15のステップS 121では、受信したコマンドが経路応答コマンドRREP_localであるか否かを判断し、“YES”であれば、さらにステップS 123で承諾か拒否かを判断する。ステップS 123で“YES”であれば、経路要求が承諾されたので、ステップS 125で、受信したポート30に向けて、経路確保確認コマンドCONFIRM_RREPを送信する。そして、ステップS 127で、ネットワーク経路応答コマンドRREP_entire待ち処理を開始する。また、ステップS 129で、自分の経路データ(図10(A))をメモリ22に生成する。ステップS 127で開始されるネットワーク経路応答待ち処理は図16に示され、コマンド処理等と並列的に実行される。なお、このネットワーク経路応答待ち処理でのタイムアウトは通常は起こらない。つまり、承諾を示す経路応答を受信して経路確保確認を送信した全てのポート30から、所定時間内にネットワーク経路応答コマンドを受信できる。

20

【0107】

ネットワーク経路応答コマンド待ち処理を開始すると、図16のステップS 141で、ネットワーク経路応答コマンド待ちタイマのカウントをスタートする。次に、ステップS 143で、タイムアウトになったか否かを判断する。また、ステップS 143で“NO”であれば、ステップS 145で、全ての葉側ポート30から経路応答コマンドRREP_localが受信されたか否かを判断する。葉側ポート30は、経路応答コマンド(承諾)を受信したポート30であり、つまり、経路確保確認コマンドを送信したポート30である。

30

【0108】

ステップS 143で“YES”であれば、またはステップS 145で“YES”であれば、ステップS 147で、ホスト側のポート30に向けて、ネットワーク経路応答コマンドRREP_entireを送信する。この場合には、全ての葉側のポート30からネットワーク経路応答コマンドを受信できていない。そこで、経路データとして、葉側のポート30からネットワーク経路応答コマンドを1度も受信していない場合には自分の経路データのみを送信し、そうでない場合には、自分の経路データとそれまでに受信した葉側の経路データとで生成した経路データ(図10(B))を送信する。そして、ステップS 149で、全てのポート30に向けて、ネットワーク経路応答待ちでのエラーを示すエラー番号Enを含むエラーコマンドErrorを送信する。

40

【0109】

また、ステップS 145で“NO”であれば、ステップS 151で、全ての葉側のポート30からネットワーク経路応答コマンドRREP_entireが受信されたか否かを判断する。ステップS 151で“NO”であれば、ステップS 143に戻り、この待ち処理を継続する。一方、ステップS 151で“YES”であれば、またはステップS 149を終了すると、このネットワーク経路応答コマンド待ち処理を終了する。

【0110】

図15に戻って、ステップS 123で“NO”であれば、つまり、経路応答が拒否であった場合には、ステップS 131で、当該ポート番号から経路応答の拒否があったことをメモリ22の所定領域に記憶する。ステップS 131またはステップS 129を終了する

50

と、このコマンド処理を終了する。なお、ステップS 1 2 1で“NO”であれば処理は図17のステップS 1 6 1へ進む。

【0111】

図17のステップS 1 6 1では、受信コマンドが経路確保確認コマンドCONFIRM__REPであったか否かを判断し、“YES”であれば、ステップS 1 6 3で、既に経路を確保しているか否かを判断する。具体的には、当該コマンドを受信したポート番号と異なるポート番号が経路確保情報記憶領域46に既に記憶されているか否かを判断する。

【0112】

ステップS 1 6 3で“NO”であれば、ステップS 1 6 5でホップ番号Hnに1を加算することによって、ホップ番号を更新してホップ番号記憶領域44に記憶する。続いて、ステップS 1 6 7で、当該経路確保確認コマンドの受信ポート30を除く全てのポート30へ向けて、経路要求コマンドREQを送信する。そして、ステップS 1 6 9で、経路応答コマンド待ち処理を開始する。この待ち処理は図18に示され、コマンド処理等と並列的に実行される。

【0113】

経路応答コマンド待ち処理を開始すると、図18のステップS 1 8 1で、経路応答コマンド待ちタイマのカウントをスタートする。次に、ステップS 1 8 3でタイムアウトになったか否かを判断する。また、ステップS 1 8 3で“NO”であれば、ステップS 1 8 5で、経路要求コマンドREQを送信した全てのポート30から経路応答コマンドREP__local（拒否）を受信したか否かを判断する。ステップS 1 8 3で“YES”であれば、またはステップS 1 8 5で“YES”であれば、ステップS 1 8 7で、自分がネットワークの末端になることをメモリ22の所定領域に記憶する。続いて、ステップS 1 8 9で、自分の経路データ（図10（A））をメモリ22の所定領域に生成する。そして、ステップS 1 9 1で、経路確保したホスト側のポート30に向けて、ネットワーク経路応答コマンドREP__entireを送信する。この場合、ステップS 1 8 9で生成した自分の経路データを送信する。

【0114】

一方、ステップS 1 8 5で“NO”であれば、ステップS 1 9 3で、経路応答コマンドREP__local（承諾）を受信されたか否かを判断する。ステップS 1 9 3で“NO”であれば、ステップS 1 8 3に戻ってこの待ち処理を継続する。また、ステップS 1 9 3で“YES”であれば、またはステップS 1 9 1を終了すると、この待ち処理を終了する。

【0115】

図17に戻って、ステップS 1 6 3で“YES”であれば、つまり、既に別の経路が確保されている場合には、そのままコマンド処理を終了する。また、ステップS 1 6 9で待ち処理を開始するとこのコマンド処理を終了する。なお、ステップS 1 6 1で“NO”であれば、処理は図19のステップS 2 0 1へ進む。

【0116】

図19のステップS 2 0 1では、受信コマンドがネットワーク経路応答コマンドREP__entireであるか否かを判定する。ステップS 2 0 1で“YES”であれば、ステップS 2 0 3で、受信したコマンドに含まれる経路データに基づいて、経路データを生成する。つまり、図10（B）に示したように、それまでに生成した経路データ（最初は自分の経路データ）に、ホップ数に関する情報、受信した葉側のポート番号、および受信した葉側の経路データを追加する。これによって、自ノードを根とするツリー構造を示すネットワーク経路データが生成される。

【0117】

続いて、ステップS 2 0 5で、全ての葉側ポート30からネットワーク経路応答コマンドREP__entireを受信したか否かを判断する。葉側ポート30は、経路応答コマンド（承諾）を受けたポート30である。ステップS 2 0 5で“NO”であれば、そのままこのコマンド処理を終了する。この場合、ネットワーク経路応答待ち処理が継続され

10

20

30

40

50

る。一方、ステップS 2 0 5で“ Y E S ”であれば、ステップS 2 0 7で、経路確保情報記憶領域4 6に記憶されたホスト側のポート3 0へ、ネットワーク経路応答コマンドR R E P _ e n t i r eを送信する。このネットワーク経路応答コマンドによって、メモリ2 2に生成された経路データをホスト側のノード1 4に送信する。ステップS 2 0 7を終了するとこのコマンド処理を終了する。なお、ステップS 2 0 1で“ N O ”であれば、処理は図2 0のステップS 2 2 1へ進む。

【 0 1 1 8 】

図2 0のステップS 2 2 1では、受信したコマンドがエラーコマンドE r r o rであるか否かを判断し、“ Y E S ”であれば、ステップS 2 2 3で、受信データからエラー番号E nを抽出してエラー番号履歴記憶領域5 0に記憶する。そして、ステップS 2 2 5で、エラー番号履歴データに基づいて、送信したことがあるエラーメッセージ(エラー番号E n)であるか否かを判断する。ステップS 2 2 5で“ Y E S ”であれば、つまり、一度送信したことがあるエラーメッセージであれば、当該エラーコマンドを送信しない。これによってメッセージのループを防ぐ。一方、ステップS 2 2 5で“ N O ”であれば、ステップS 2 2 7で、残りのポート3 0に向けて当該エラーコマンドを送信する。ステップS 2 2 7を終了するとこのコマンド処理を終了する。なお、ステップS 2 2 1で“ N O ”であれば、処理は図2 1のステップS 2 4 1へ進む。

【 0 1 1 9 】

図2 1のステップS 2 4 1では、受信したコマンドが識別番号割当コマンドであるか否かを判断する。ステップS 2 4 1で“ Y E S ”であれば、ステップS 2 4 3で、その世代番号における識別番号を既に持っているか否かを判断する。ステップS 2 4 3で“ N O ”であれば、ステップS 2 4 5で、受信コマンドに含まれる識別番号をたとえば世代番号に関連付けて識別番号記憶領域4 8に記憶する。そして、ステップS 2 4 7で、センサデータの送信処理を開始する。センサデータの送信処理は、後述する図2 4に示され、コマンド処理等と並列的に実行される。一方、ステップS 2 4 3で“ Y E S ”であれば、ステップS 2 4 9で全てのポート3 0に向けて、識別番号割当におけるエラーを示すエラー番号E nを含むエラーコマンドE r r o rを送信する。ステップS 2 4 7でセンサデータ送信処理を開始し、またはステップS 2 4 9を終了すると、このコマンド処理を終了する。なお、ステップS 2 4 1で“ N O ”であれば、処理は図2 2のステップS 2 6 1へ進む。

【 0 1 2 0 】

図2 2のステップS 2 6 1では、受信コマンドがノード中継コマンドR E L A Y _ t o _ N o d eであるか否かを判断する。ステップS 2 6 1で“ Y E S ”であれば、ステップS 2 6 3で、当該コマンドに含まれるデータD a t aがコマンドであるか否かを判断する。ステップS 2 6 3で“ Y E S ”であれば、ステップS 2 6 5で、指定されたポート番号のポート3 0へ向けて、当該コマンドを送信する。

【 0 1 2 1 】

一方、ステップS 2 6 3で“ N O ”であれば、ステップS 2 6 7で、指定ポート番号がローカルホストを示すか否かを判断し、“ N O ”であれば、ステップS 2 6 9で、指定ポート番号のポート3 0に向けて、当該データをノード中継コマンドR E L A Y _ t o _ N o d eで送信する。

【 0 1 2 2 】

また、ステップS 2 6 7で“ Y E S ”であれば、当該データはこのノード1 4宛のデータであるので、ステップS 2 7 1で、当該データをメモリ2 2の所定領域に記憶する。このデータが隣接センサ出力データである場合には、隣接センサ出力記憶領域5 4に記憶する。ステップS 2 6 5、S 2 6 9またはS 2 7 1を終了すると、このコマンド処理を終了する。なお、ステップS 2 6 1で“ N O ”であれば、処理は図2 3のステップS 2 8 1へ進む。

【 0 1 2 3 】

図2 3のステップS 2 8 1では、受信コマンドがホスト中継コマンドR E L A Y _ t o _ H o s tであるか否かを判断する。ステップS 2 8 1で“ Y E S ”であれば、ステップ

10

20

30

40

50

S 2 8 3で優先順位判定処理を実行する。この処理によって、優先度の高いセンサデータが受信された場合には、当該データを優先的に送信できるようにバッファの空きを調整する。この優先順位判定処理の動作は後述する図 2 8 に示される。そして、ステップ S 2 8 5では、ホスト側のポート 3 0 に向けて、ホスト中継コマンド R E L A Y _ t o _ H o s t を用いてデータを送信する。これによって、優先度の高いデータから優先的に送信される。ステップ S 2 8 5 を終了するとこのコマンド処理を終了する。

【 0 1 2 4 】

また、ステップ S 2 8 1 で “ N O ” であれば、つまり、受信したコマンドが未定義コマンドである場合には、ステップ S 2 8 7 で、未定義コマンドの受信を示すエラー番号 E n 10
を含むエラーコマンド E r r o r を全てのポート 3 0 に向けて送信し、このコマンド処理を終了する。

【 0 1 2 5 】

図 2 4 には、図 2 1 のステップ S 2 4 7 で開始されるセンサデータの送信処理の動作の一例が示される。センサデータ送信処理を開始すると、ステップ S 3 0 1 で、識別番号記憶領域 4 8 のデータを参照して、識別番号を割り当てられているか否かを判断する。ステップ S 3 0 1 で “ Y E S ” であれば、ステップ S 3 0 3 で、センサ出力履歴記憶領域 5 4 を参照して、今回のセンサ出力データを取得済みであるか否かを判断する。ステップ S 3 0 3 で “ N O ” であれば、センサ出力が取得されるのを待つ。

【 0 1 2 6 】

ステップ S 3 0 3 で “ Y E S ” であれば、ステップ S 3 0 5 でデータ圧縮処理を実行する。この実施例では、空間方向の圧縮、時間方向の圧縮、ノイズレベルの圧縮の 3 つの処理が実行される。空間方向のデータ圧縮処理は後述する図 2 5 に示される。時間方向のデータ圧縮処理は後述する図 2 6 に示される。また、ノイズレベルのデータ圧縮処理は後述する図 2 7 に示される。 20

【 0 1 2 7 】

続いて、ステップ S 3 0 7 ではセンサデータの送信が必要であるか否かを判断する。具体的には、データ圧縮処理でマスクしない（つまり、送信が必要）と判断されたセンサデータが存在するか否かを判断する。ステップ S 3 0 7 で “ Y E S ” であれば、ステップ S 3 0 9 で、優先度情報に基づいて、送信データの優先度を設定する。たとえば、優先度情報として痛い感覚を示すデータや撫でられていることを示すデータ等のようにセンサ出力データの値によって決まるデータが設定される場合には、センサ出力データの値に基づいて、優先度が設定される。また、優先度情報として危ないことを示すデータ等のようなセンサ位置もしくはセンサ I D によって決まるデータが設定される場合には、送信の必要なセンサ出力データのセンサ位置もしくはセンサ I D に基づいて、優先度が設定される。送信するセンサデータには、センサ I D だけでなくさらに優先度に関するラベルが付与される。たとえば、痛い感覚を示すラベル、撫でられていることを示すラベル、危ないことを示すラベル、あるいは高優先度を示すラベル等が付与される。 30

【 0 1 2 8 】

そして、ステップ S 3 1 1 で、ホスト側のポート 3 0 に向けて、ホスト中継コマンド R E L A Y _ t o _ H o s t によってセンサデータ（触覚情報）を送信する。このホスト中継コマンドで送信するデータにはさらに当該ノード 1 4 の識別番号が付与される。ステップ S 3 1 1 を終了するとこの送信処理を終了する。一方、ステップ S 3 0 7 で “ N O ” 10
であれば、つまり、このノード 1 4 に所属する全てのセンサ 1 6 のセンサデータをマスクする場合には、そのままこの送信処理を終了する。また、ステップ S 3 0 1 で “ N O ” である場合もそのままこの送信処理を終了する。

【 0 1 2 9 】

図 2 5 には、図 2 4 のステップ S 3 0 5 の処理のうち空間方向のデータ圧縮処理の動作の一例が示される。まず、データを圧縮しようとする 1 つのセンサ I D を決めて、当該センサデータに対して処理を実行する。ステップ S 3 2 1 では、隣接センサ出力データを取得済みであるか否かを判断する。具体的には、隣接センサ位置情報に記憶される他のノード 50

ド 1 4 に所属する隣接センサのセンサ出力データが隣接センサ出力記憶領域 5 6 に記憶されているか否かを判断する。ステップ S 3 2 1 で “ N O ” であれば、隣接センサ出力データの取得を待つ。

【 0 1 3 0 】

ステップ S 3 2 1 で “ Y E S ” であれば、ステップ S 3 2 3 で、隣接センサデータからマスク領域を算出する。マスク領域は、たとえば図 5 に示したように隣接センサデータの値と係数 k 等に基づいて算出される。そして、ステップ S 3 2 5 で、処理対象のセンサのデータの値がマスク領域内であるか否かを判定する。ステップ S 3 2 5 で “ Y E S ” であれば、ステップ S 3 2 7 で当該処理対象のセンサ ID のセンサデータを送信しないすなわちマスクすることをメモリ 2 2 に記憶する。一方、ステップ S 3 2 5 で “ N O ” であれば、ステップ S 3 2 9 で当該センサ ID のセンサデータは送信が必要でありマスクしないことをメモリ 2 2 に記憶する。メモリ 2 2 にはセンサ ID ごとにマスクするかしなかを示すデータが記憶される。

10

【 0 1 3 1 】

続いてステップ S 3 3 1 で全てのセンサ ID について処理を完了したか否かを判断し、“ N O ” であれば、ステップ S 3 2 1 へ戻って、次のセンサ ID のセンサデータに対して処理を実行する。ステップ S 3 3 1 で “ Y E S ” であれば、この空間方向のデータ圧縮処理を終了する。

【 0 1 3 2 】

図 2 6 には、図 2 4 のステップ S 3 0 5 の処理のうち時間方向のデータ圧縮処理の動作の一例が示される。空間方向の場合と同様に、まず、圧縮処理対象のセンサ ID を決定し、当該センサデータに対して処理を実行する。ステップ S 3 4 1 では、センサ出力履歴データからマスク領域を算出する。マスク領域は、たとえば図 6 に示したようにセンサ出力データの履歴と係数 k_t とに基づいて算出される。そして、ステップ S 3 4 3 で、処理対象のデータの値がマスク領域内であるか否かを判定する。ステップ S 3 4 3 で “ Y E S ” であれば、ステップ S 3 4 5 で当該処理対象のセンサ ID のセンサデータを送信しないすなわちマスクすることをメモリ 2 2 に記憶する。一方、ステップ S 3 4 3 で “ N O ” であれば、ステップ S 3 4 7 で当該センサ ID のセンサデータは送信が必要でありマスクしないことをメモリ 2 2 に記憶する。

20

【 0 1 3 3 】

続いて、ステップ S 3 4 9 で全てのセンサ ID について処理を実行したか否かを判断し、“ N O ” であれば、ステップ S 3 4 1 に戻って、次のセンサ ID のセンサデータに対して処理を実行する。ステップ S 3 4 9 で “ Y E S ” であれば、この時間方向のデータ圧縮処理を終了する。

30

【 0 1 3 4 】

図 2 7 には、図 2 4 のステップ S 3 0 5 の処理のうちノイズレベルのデータ圧縮処理の動作の一例が示される。まず、ステップ S 3 6 1 で、ノイズレベル $N(N_t)$ が設定されているか否かをノイズレベル情報記憶領域 6 0 に基づいて判断する。ステップ S 3 6 1 で “ N O ” であれば、そのままこの圧縮処理を終了する。

【 0 1 3 5 】

一方、ステップ S 3 6 1 で “ Y E S ” であれば、圧縮処理対象のセンサ ID を決定し、当該センサデータに対して処理を実行する。具体的には、ステップ S 3 6 3 でセンサデータの変化量を算出する。たとえば、図 7 に示したように、隣接センサ $i + 1$ と隣接センサ $i - 1$ の隣接センサ出力データに基づいて、隣接センサに対する当該センサの空間方向の変化量を算出する。また、センサ出力データの履歴から当該センサの時間方向の変化量を算出する。

40

【 0 1 3 6 】

そして、ステップ S 3 6 5 で、それぞれの変化量がそれぞれのノイズレベル以下であるか否かを判定する。ステップ S 3 6 5 で “ Y E S ” であれば、つまり、空間および時間のいずれか一方で変化量がノイズレベル以下であれば、ステップ S 3 6 7 で、当該センサ I

50

Dのセンサデータを送信しない(マスクする)ことをメモリ22に記憶する。一方、ステップS365で“NO”であれば、当該センサIDのセンサデータは送信が必要である(マスクされない)ことをメモリ22に記憶する。

【0137】

続いて、ステップS371で、全てのセンサIDについて処理を実行したか否かを判断し、“NO”であれば、ステップS363に戻って、次のセンサIDのセンサデータに対して処理を実行する。ステップS371で“YES”であれば、このノイズレベルのデータ圧縮処理を終了する。

【0138】

図28には、図23のステップS283の優先順位判定処理の動作の一例が示される。ステップS381で、ホスト中継コマンドで受信したデータは、次に送るデータよりも優先度の高いデータであるか否かを判断する。具体的には、センサデータに高優先度のラベル(たとえば、痛い感覚、危ない等)が付与されているか否かを判断するとともに、次に送ろうとしているデータ、すなわち、ホスト側ポートの送信バッファのデータの優先度に高優先度のラベルが設定されていないかどうかを判断する。あるいは、センサデータを受信するたびに、当該受信したセンサデータの優先度を優先度情報に基づいて判定するようにしてもよい。ステップS381で“NO”であれば、そのままこの優先順位判定処理を終了し、図23のステップS285へ進む。

10

【0139】

一方、ステップS381で“YES”であれば、ステップS383で、ホスト側ポート30の転送データバッファ28の送信バッファに、当該ホスト中継コマンドを送信する余裕(記憶領域)が存在するか否かを判断する。ステップS383で“NO”であれば、ステップS385で、既書き込まれている低優先度のデータを送信バッファからメモリ22へ戻す。また、ステップS383で“YES”であれば、そのままステップS387へ進む。そして、ステップS387で、ホスト側ポートの送信バッファを、当該高優先度のホスト中継コマンドの送信に必要なデータ分だけ空ける。ステップS387を終了すると、処理は図23のステップS285へ進む。こうして、当該高優先度のセンサデータが先に送信されることとなる。したがって、ホストコンピュータ12は、高優先度の触覚情報を先に受信することができ、たとえば緊急を要する触覚情報を逸早く認識して素早く対応することが可能になる。

20

30

【0140】

経路確保する際のノード14間通信の処理の流れを図29および図30に示す。図29には、たとえばノード1がノード2との経路確保に成功する場合のタイムチャートが示される。

【0141】

ノード1は、ステップS167で経路要求コマンドRREQをノード2に送信し、ステップS169で経路応答待ち処理を開始する。

【0142】

ノード2は、ノード1からのコマンドを受信して、ステップS73で経路要求コマンドRREQを受信したと判断し、ステップS75で世代番号Gnおよびホップ番号Hnを取得する。続くステップS77で、ノード2は世代番号Gnが同じでないことを確認し、ステップS79でホスト側の経路確保を行い、ステップS81で受信した世代番号Gnを記憶する。つまり、ノード2は当該世代番号Gnを含む経路要求コマンドRREQを初めて受信したので、その送信元であるノード1にホスト側経路を確保する。そして、ステップS83で、ノード2は経路応答コマンドRREP_local(承諾)をノード1に送信し、ステップS85で経路確保確認コマンドCONFIRM_RREP待ち処理を開始する。

40

【0143】

ノード1は、ノード2からのコマンドを受信して、ステップS121で経路応答コマンドRREP_localを受信したと判断し、さらにステップS123で承諾を確認する

50

。そして、ステップS 1 2 5で経路確保確認コマンドCONFIRM__RREPをノード2に送信し、ステップS 1 2 7でネットワーク経路応答待ち処理を開始する。

【0144】

ノード2は、ノード1からのコマンドを受信して、ステップS 1 6 1で経路確保確認コマンドCONFIRM__RREPを受信したと判断し、ステップS 1 6 3で未だ経路が確保されていないことを確認する。そして、ステップS 1 6 5でホップ番号Hnを更新し、ステップS 1 6 7で経路要求コマンドREQをノード1の接続されたポートを除く全てのポート(図29の例ではノード3に接続されるポートのみ)に送信し、ステップS 1 6 9で経路応答待ち処理を開始する。以降のノード2とノード3との間の通信は、経路確保に成功する場合には、図29のノード1とノード2との間の通信と同様に進むこととなる

10

【0145】

一方、図30には、たとえばノード1がノード2との経路確保に失敗する場合のタイムチャートが示される。ノード1は、ステップS 1 6 7で経路要求コマンドREQをノード2に送信し、ステップS 1 6 9で経路応答待ち処理を開始する。

【0146】

ノード2は、ノード1からのコマンドを受信して、ステップS 7 3で経路要求コマンドREQを受信したと判断し、ステップS 7 5で世代番号Gnおよびホップ番号Hnを取得する。続くステップS 7 7で、ノード2は世代番号Gnが同じであることを確認する。つまり、ノード2は当該世代番号Gnを含む経路要求コマンドREQを既に他のノードから受信しているため、ノード1の経路要求を拒否する。したがって、ノード2は、ステップS 8 7で、経路応答コマンドREP__local(拒否)をノード1に送信する。

20

【0147】

ノード1は、ノード2からのコマンドを受信して、ステップS 1 2 1で経路応答コマンドREP__localを受信したと判断する。さらにステップS 1 2 3で拒否を確認して、ステップS 1 3 1でノード2の接続されたポートから経路応答コマンド(拒否)を受信したことを記憶する。

【0148】

この後、ノード1が、経路応答コマンド(承諾)を受信することなくタイムアウトになった場合、あるいは経路要求を送信した全てのポート30から経路応答コマンド(拒否)を受信した場合には、図18に示したように、ノード1は、自分がネットワークの端末になると判定し、自分の経路データを生成して、ネットワーク経路応答コマンドをホスト側経路であるホスト側のポート30に向けて送信する。したがって、このノード1の経路上のホスト側に位置するノードは、図19に示したように、ネットワーク経路応答コマンドを受信して、受信した経路データと自分の経路データとを結合してネットワーク経路データを生成する。そして、当該ノードは、経路応答の承諾を受けた全てのポート30からネットワーク経路応答コマンドを受け取ったと判定される場合には、生成したネットワーク経路データをホスト側のポート30に向けて送信する。このようにして、ホストコンピュータ12にネットワーク全体の経路を示す経路データが送信されることとなる。

30

【0149】

この実施例によれば、ノード14間の相互接続による通信によって、自己組織化可能で頑丈かつ簡単な情報伝達経路を構築することができる。そして、ノード14の境界で隣接する触覚センサ16の出力をノード14間で相互に通信し、各ノード14において触覚センサ出力データの空間方向圧縮、時間方向圧縮、あるいはノイズレベル圧縮などを行うことができるので、ホストコンピュータ12に転送するセンサデータ量を減少させることができる。したがって、転送のボトルネックが発生するのを回避することができる。また、ホストコンピュータ12での情報処理の負荷を軽減することができる。

40

【0150】

発明者等は上述の実施例のセンサネットワーク10におけるデータ圧縮方法の有効性を検証した。その実験では、複数の触覚センサ16は図31に示すように配置された。各触

50

覚センサエレメント (P V D F) 1 6 は、 3 0 m m × 3 0 m m の大きさを有し、 約 5 m m 間隔程度で 1 m m のウレタンスポンジ上に並べられ、 厚さ 1 m m のシリコンシートで覆われている。 センサデータはノードから P C に読み込まれた。 サンプルングは 1 0 0 H z で行い、 測定レンジは - 5 1 2 から 5 1 2 であり 0 は基準点である。

【 0 1 5 1 】

センサデバイスの中央を「手の平で 1 から 2 秒間隔で軽く押さえた」場合と、「腕で 1 から 2 秒間隔で軽く押さえた」場合の 2 つの触り方で得られたセンサデータについて、 上述の空間方向の圧縮方法を適用した結果が図 3 2 および図 3 3 にそれぞれ示される。 図 3 1 のセンサ 7 のセンサデータについての結果を示して、 各図の (A) は圧縮前の反応 (生データ) であり、 (B) は圧縮法の適応結果である。 横軸は時間、 縦軸はセンサ反応値である。 マスク範囲を決めるパラメータ k の値は 1 に設定され、 ノイズレベル N の値は 4 に設定された。 約 1 0 秒間のデータで圧縮率 (元のデータに対するマスクされたデータの割合) は、 図 3 2 に示す手の平の場合で 9 1 . 5 % となり、 図 3 3 に示す腕の場合で 8 3 . 4 % となった。 このように、 センサ出力の 8 ~ 9 割が圧縮されるのは、 周囲のセンサ 1 6 が同等のセンサ値を出力しているからである。 全体としてホストコンピュータ 1 2 に送信される情報は、 触覚出力の変化領域の輪郭ということになる。

【 0 1 5 2 】

また、 図示は省略するが、 上述の時間方向の外挿による圧縮方法を同じデータに適応したところ、 圧縮率は 1 0 ~ 4 0 % となった。 この 1 次線形外挿による圧縮はノイズ低減に効果的である。

【 0 1 5 3 】

なお、 上述の実施例では、 ホストコンピュータ 1 2 が各ノード 1 4 に識別番号を送信したことに応じて各ノード 1 4 は触覚情報をホスト中継コマンドを用いてホスト宛に送信するようにしていた。 しかし、 他の実施例では、 ネットワーク経路応答コマンドを用いて、 経路データとともにセンサデータを送信するようにしてもよい。 つまり、 図 1 0 (A) に示したように、 自分の経路データは、 任意のデータを登録できる `anything` 領域を含むので、 各ノード 1 4 は、 この `anything` 領域に、 マスクしない触覚情報を書き込んだ経路データを生成して、 当該データをネットワーク経路応答コマンドで送信するようにしてよい。

【 0 1 5 4 】

この他の実施例では、 ホストコンピュータ 1 2 の送信する経路確保確認コマンドが触覚情報送信要求を兼ねることとなり、 ホストコンピュータ 1 2 は、 図 1 1 のステップ S 1 1 でネットワーク経路応答コマンドを受信することによって、 ステップ S 1 3 で経路データとともに触覚情報を取得することができる。

【 0 1 5 5 】

また、 各ノード 1 4 は、 たとえば図 1 5 のステップ S 1 2 9 または図 1 8 のステップ S 1 8 9 で自分の経路データを生成する際に、 図 2 4 のステップ S 3 0 3 の判定、 ステップ S 3 0 5 のデータ圧縮処理およびステップ S 3 0 7 の判定を実行して、 マスクしない触覚情報を自分の経路データの `anything` 領域に登録する。 そして、 図 1 8 のステップ S 1 9 1 または図 1 9 のステップ S 2 0 7 で、 ネットワーク経路応答コマンドを用いて、 触覚情報を含む経路データを送信する。

【 0 1 5 6 】

また、 上述の各実施例では、 ホスト宛に送信される触覚情報はセンサ出力値であったが、 センサ出力値に基づくラベルデータのみを触覚情報として送信してもよい。 たとえば、 各ノード 1 4 は、 上述の優先度の設定 (ステップ S 3 0 9) と同様にして、 メモリ 2 2 に記憶されたセンサ出力値 (またはセンサ出力値の経時変化) とラベルとを関連付けたテーブルデータに基づいて、 センサ出力データに対して、 痛い感覚を示すデータ、 撫でられていることを示すデータ、 あるいは叩かれていることを示すデータといったラベリングを行う。 そして、 ノード 1 4 は触覚情報としてそのラベルデータのみを送信する。 ホストコンピュータ 1 2 は触覚情報をラベルデータで記憶する。

【図面の簡単な説明】

【0157】

【図1】この発明の一実施例の皮膚センサネットワークの概要を示す図解図である。

【図2】ノードの電氣的構成の一例を示すブロック図である。

【図3】皮膚とそこに埋め込まれる触覚センサエレメントの一部を示す図解図である。

【図4】図1実施例の経路確保の概要を示す図解図であり、(A)はホストコンピュータからの経路確保要求の伝達を示し、(B)は確立された経路の一例を示し、(C)は断線時に再構築された経路の一例を示す。

【図5】センサ情報の空間での圧縮方法の概要を示す図解図である。

【図6】センサ情報の時間軸方向の圧縮方法の概要を示す図解図である。 10

【図7】ノイズ除去の概要を示す図解図である。

【図8】ノードのメモリのメモリマップの一例の一部を示す図解図である。

【図9】隣接センサ位置情報の内容の一例を示す図解図である。

【図10】ノードで生成される経路データの内容の一例を示す図解図であり、図10(A)は自分の経路データを示し、図10(B)はホスト側へ送信する経路データを示す。

【図11】図1実施例のホストコンピュータの動作の一例を示すフロー図である。

【図12】図1実施例の各ノードの動作の一例を示すフロー図である。

【図13】図12のコマンド処理の動作の一例の一部を示すフロー図である。

【図14】図13の経路確保確認コマンド待ち処理の動作の一例を示すフロー図である。

【図15】図13の続きの一部を示すフロー図である。 20

【図16】図15のネットワーク経路応答コマンド待ち処理の動作の一例を示すフロー図である。

【図17】図15の続きの一部を示すフロー図である。

【図18】図17の経路要求応答待ち処理の動作の一例を示すフロー図である。

【図19】図17の続きの一部を示すフロー図である。

【図20】図19の続きの一部を示すフロー図である。

【図21】図20の続きの一部を示すフロー図である。

【図22】図21の続きの一部を示すフロー図である。

【図23】図22の続きの一部を示すフロー図である。

【図24】図21のセンサデータの送信処理の動作の一例を示すフロー図である。 30

【図25】図24のデータ圧縮処理(空間方向)の動作の一例を示すフロー図である。

【図26】図24のデータ圧縮処理(時間方向)の動作の一例を示すフロー図である。

【図27】図24のデータ圧縮処理(ノイズレベル)の動作の一例を示すフロー図である。

【図28】図23の優先順位判定処理の動作の一例を示すフロー図である。

【図29】ルーティングのためのノード間通信によってノード1が経路確保する場合の動作の一例を示すタイムチャートである。

【図30】ルーティングのためのノード間通信によってノード1が経路確保に失敗する場合の動作の一例を示すタイムチャートである。

【図31】実験における触覚センサの配置を示す図解図である。 40

【図32】実験において手の平で軽く押さえたときの7番のセンサのセンサデータの一例を示す図解図であり、(A)は圧縮前のオリジナルデータを示し、(B)は圧縮後のデータを示す。

【図33】実験において腕で軽く押さえたときの7番のセンサのセンサデータの一例を示す図解図であり、(A)は圧縮前のオリジナルデータを示し、(B)は圧縮後のデータを示す。

【符号の説明】

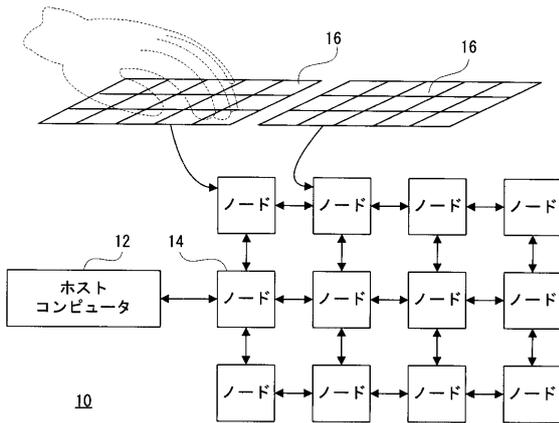
【0158】

10 ...皮膚センサネットワーク

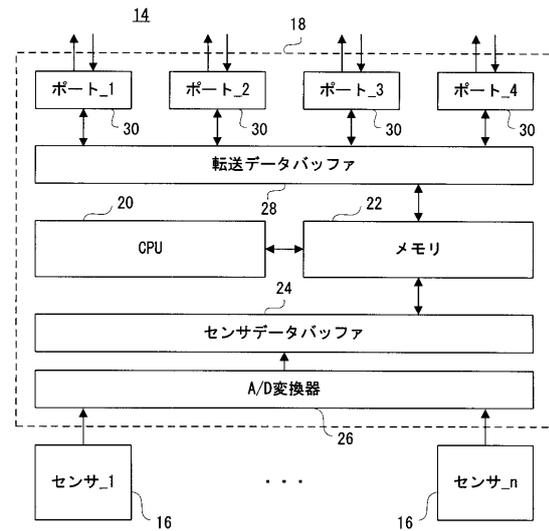
12 ...ホストコンピュータ 50

- 1 4 ... ノード
- 1 6 ... 触覚センサ
- 2 0 ... C P U
- 2 2 ... メモリ
- 2 6 ... A / D 変換器
- 3 0 ... 通信ポート

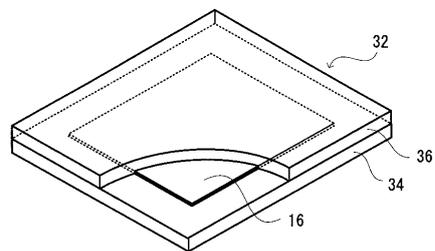
【 図 1 】



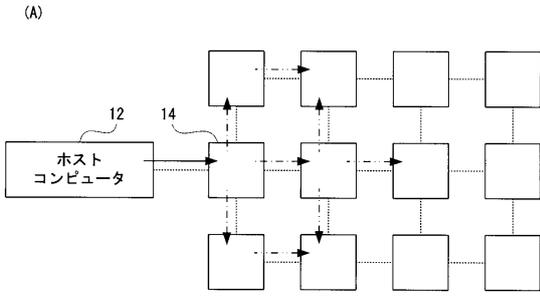
【 図 2 】



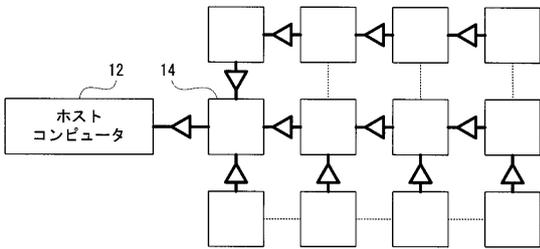
【 図 3 】



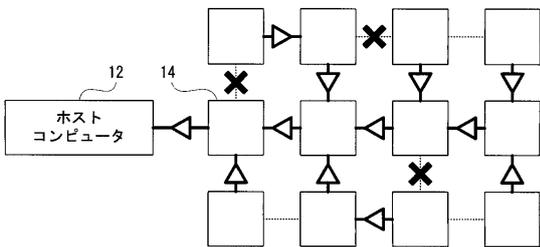
【図 4】



(B)



(C)



【図 8】

ノードのメモリマップ

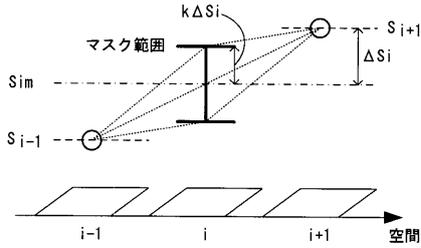
世代番号 G_n (初期値0)	42
ホップ番号 H_n	44
経路確保情報(ホスト側ポート番号)	46
識別番号	48
エラー番号 E_n 履歴	50
隣接センサ位置情報	52
センサ出力履歴	54
隣接センサ出力	56
優先度情報	58
ノイズレベル情報	60

【図 9】

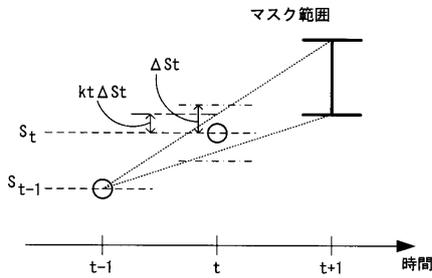
隣接センサ位置情報

センサID	隣接センサの所属ノード	隣接センサID

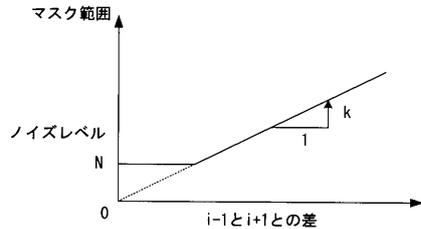
【図 5】



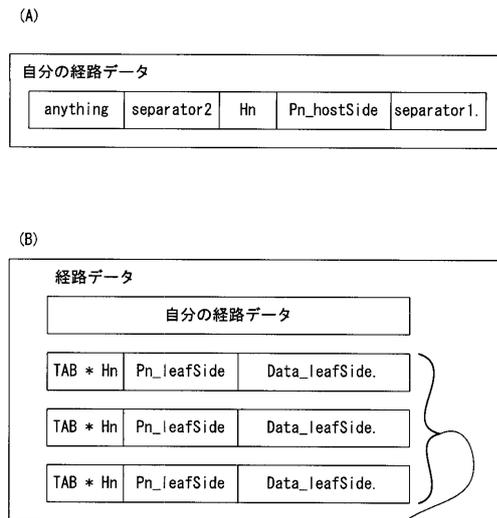
【図 6】



【図 7】

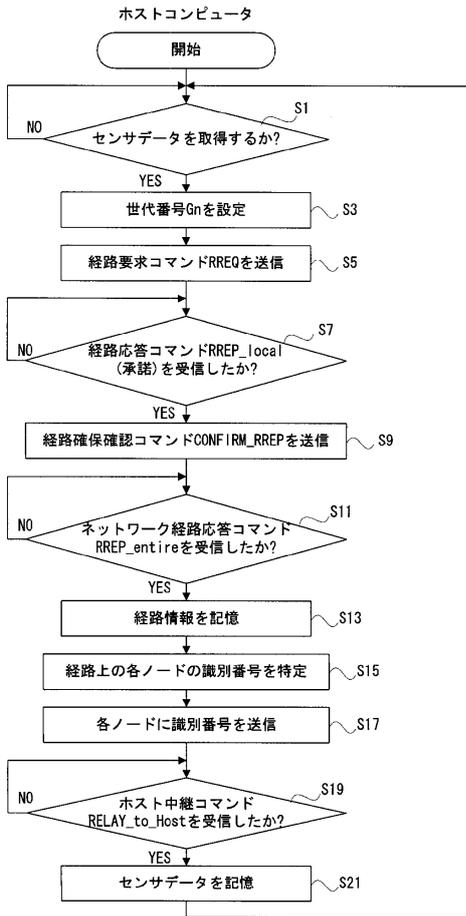


【図 10】

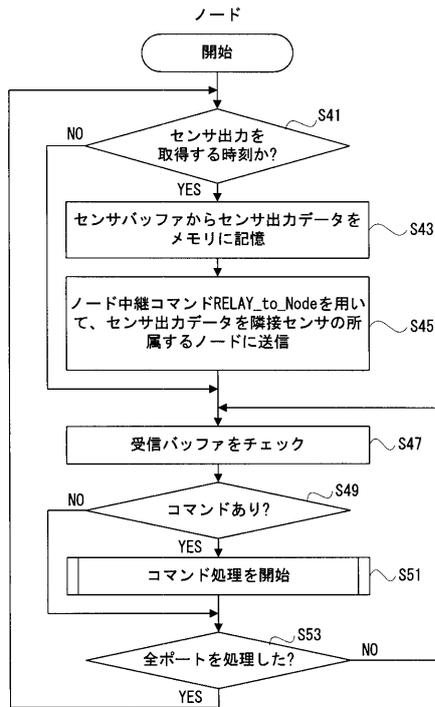


葉側ポートからの受信ごとに追加される

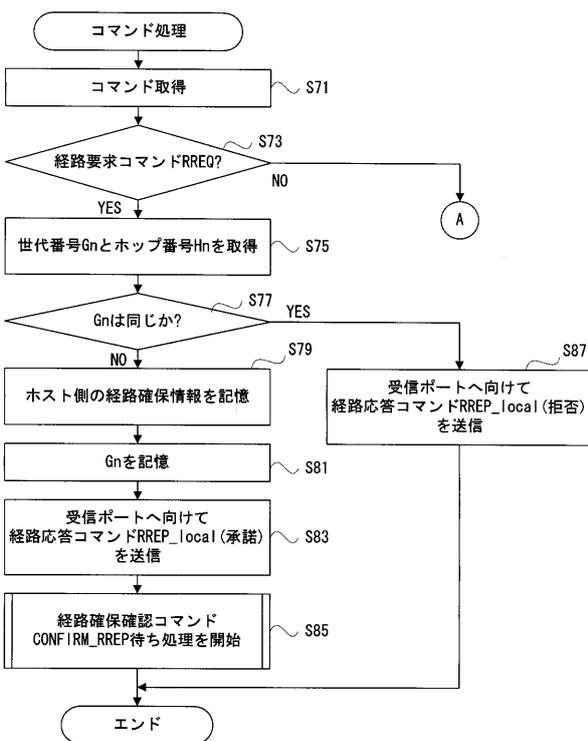
【図11】



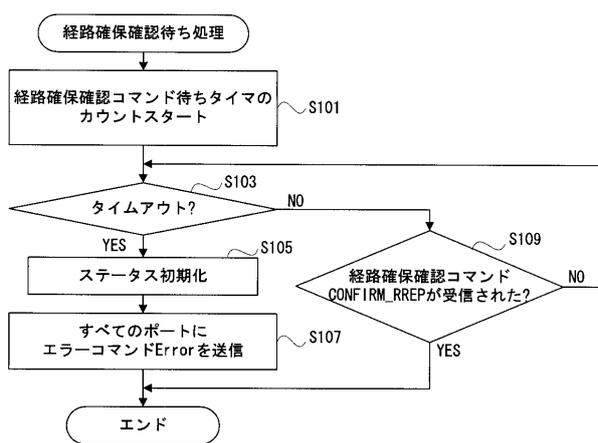
【図12】



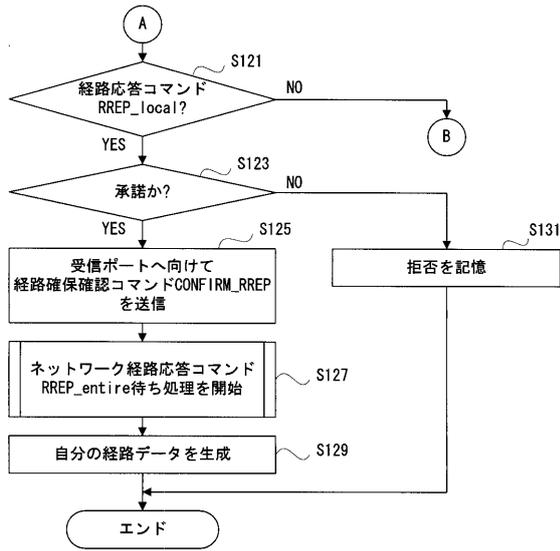
【図13】



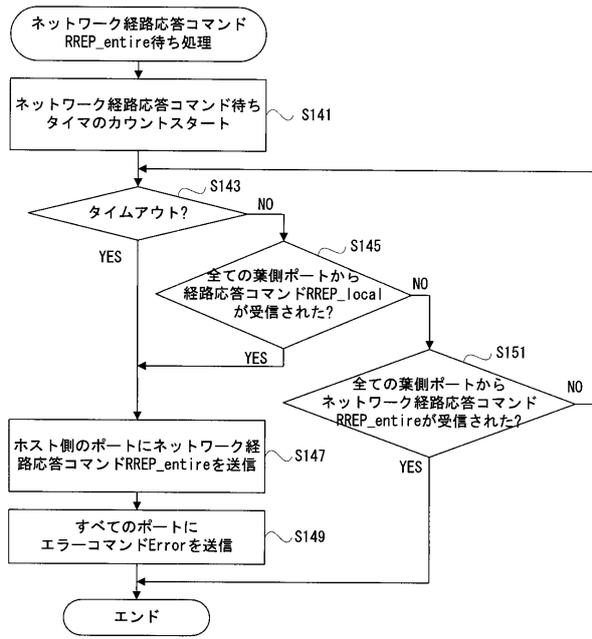
【図14】



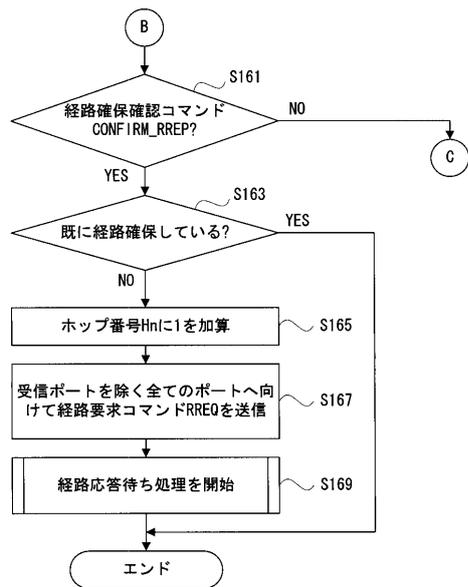
【図15】



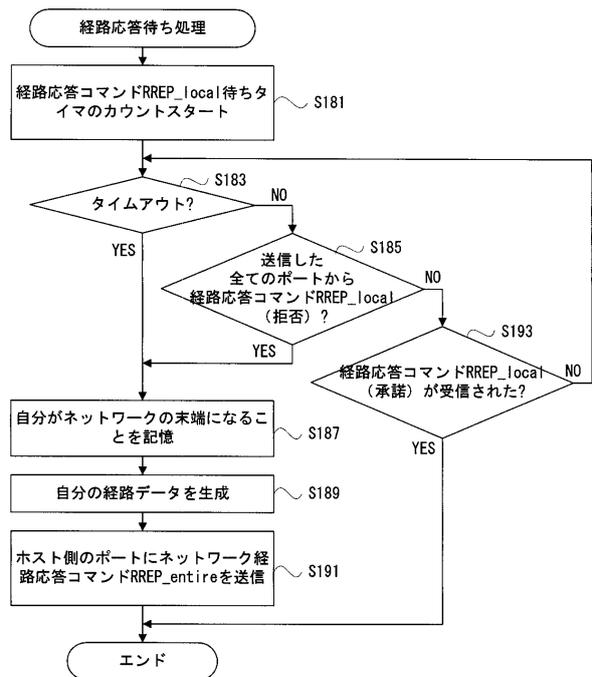
【図16】



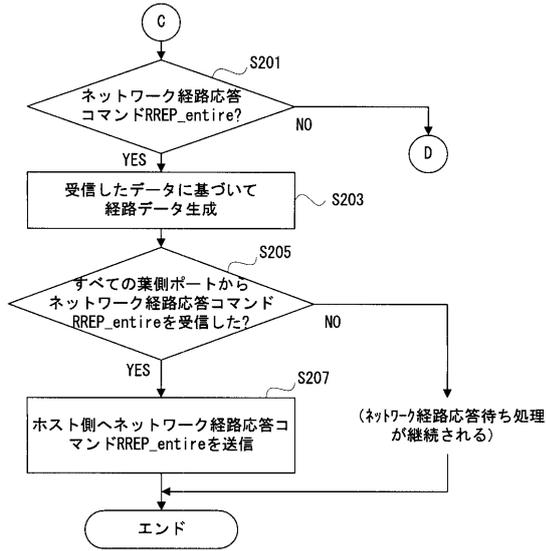
【図17】



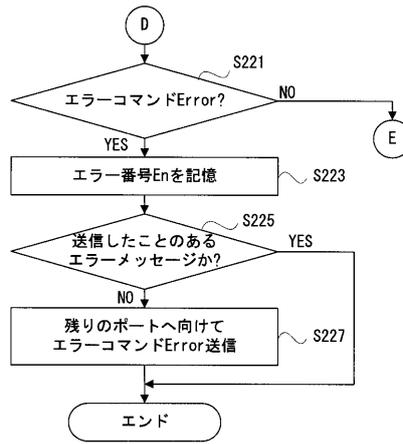
【図18】



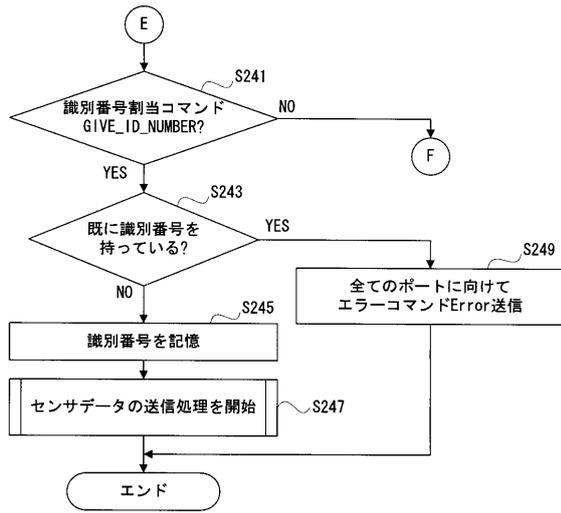
【図19】



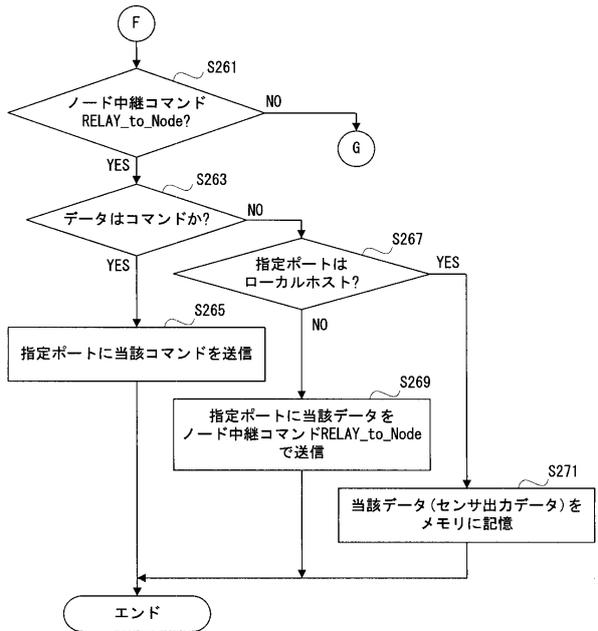
【図20】



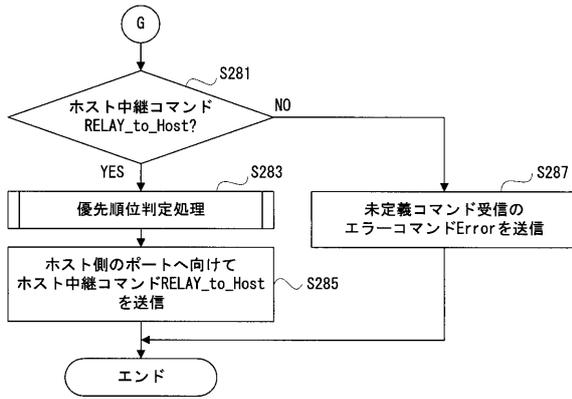
【図21】



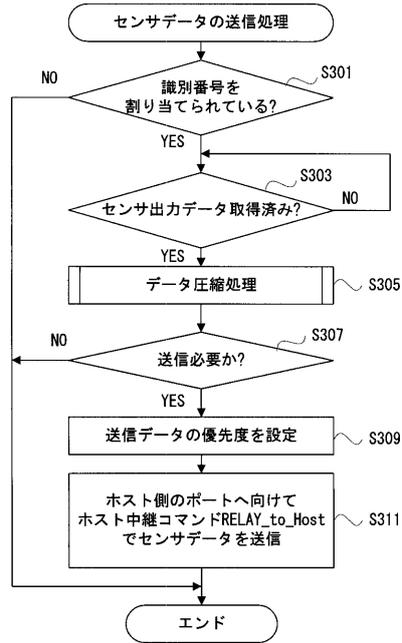
【図22】



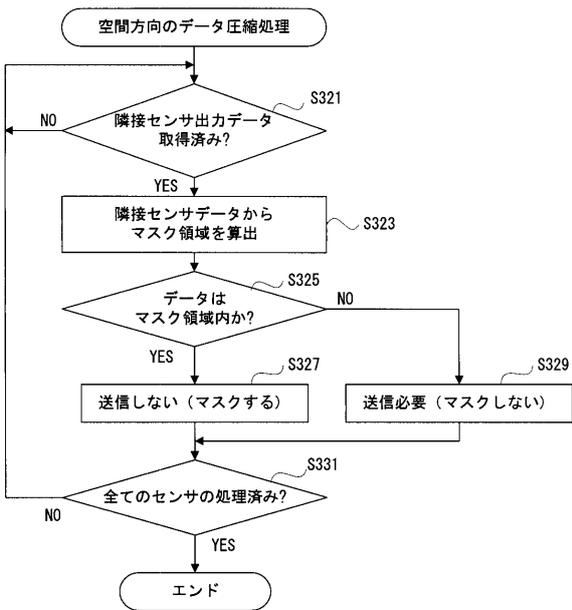
【図23】



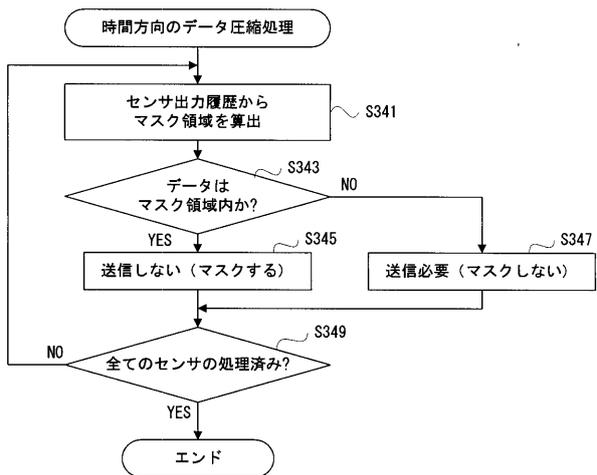
【図24】



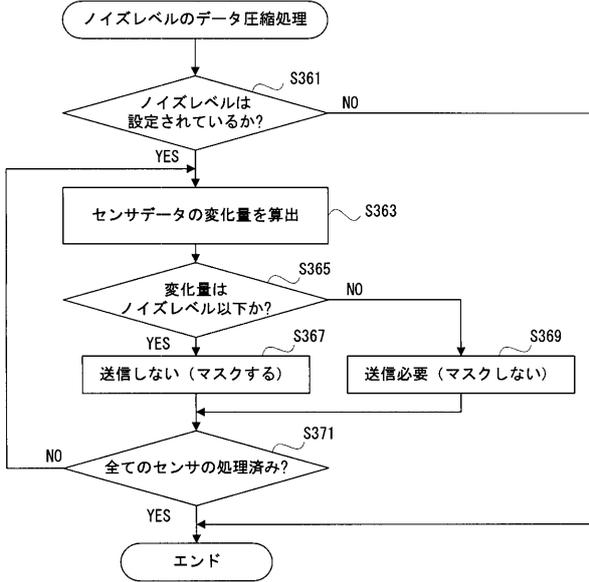
【図25】



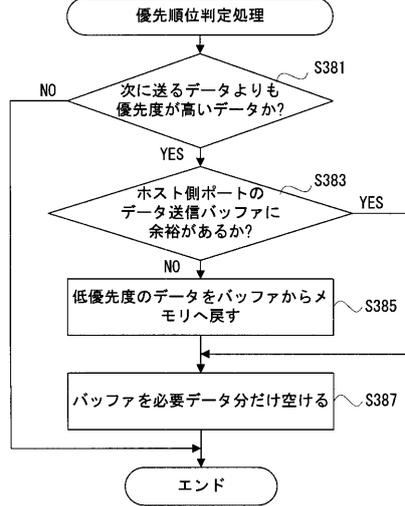
【図26】



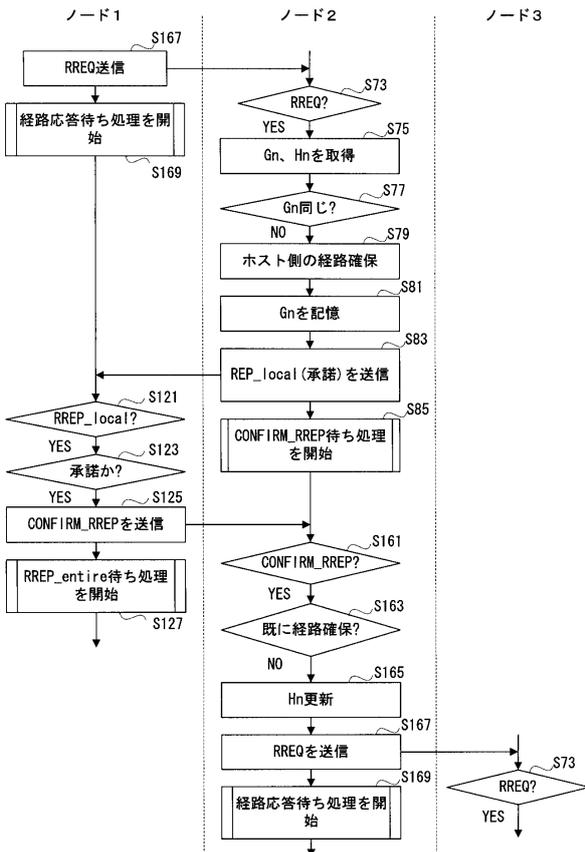
【図27】



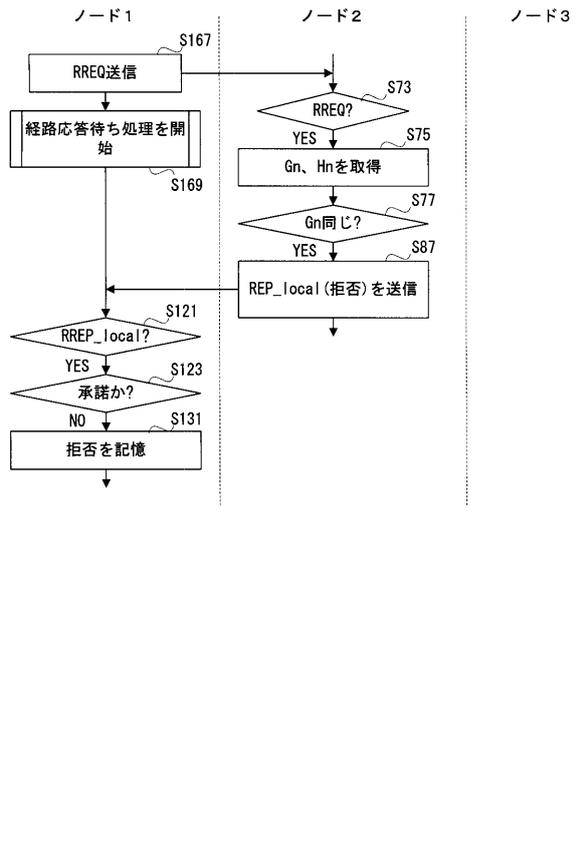
【図28】



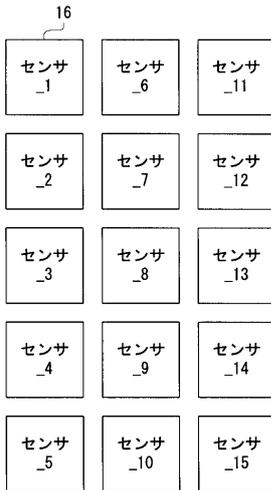
【図29】



【図30】

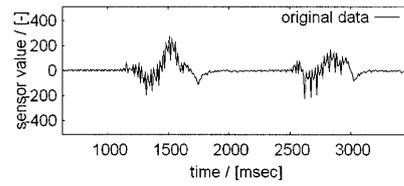


【図 3 1】

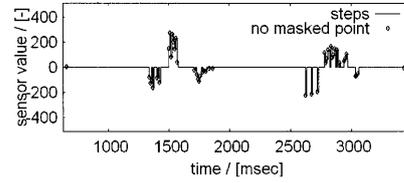


【図 3 2】

(A) 圧縮前

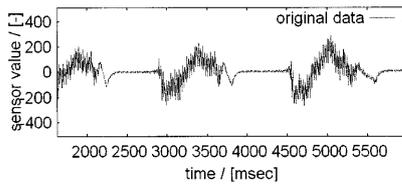


(B) 圧縮後

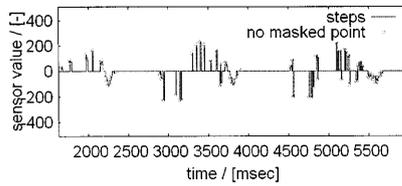


【図 3 3】

(A) 圧縮前



(B) 圧縮後



フロントページの続き

(56)参考文献 特表2005-523646(JP,A)
特開2002-116101(JP,A)
特開2003-123177(JP,A)
特開2005-39831(JP,A)
特開2004-283975(JP,A)

(58)調査した分野(Int.Cl., DB名)

H04L 12

B25J 19