(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第3563787号

(P3563787)

(45)発行日　平成16年9月8日(2004.9.8)　(24)登録日　平成16年6月11日(2004.6.11)

(51)Int.Cl.[7]	F I		
H04N 5/232	H04N	5/232	B
G06F 13/00	G06F	13/00	357Z
H04N 7/15	H04N	7/15	610

請求項の数 7　(全 13 頁)

(21)出願番号　特願平6-296681
(22)出願日　平成6年11月30日(1994.11.30)
(65)公開番号　特開平8-163422
(43)公開日　平成8年6月21日(1996.6.21)
審査請求日　平成13年10月12日(2001.10.12)

前置審査

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100090538
弁理士　西山　恵三
(74)代理人　100096965
弁理士　内尾　裕一
(72)発明者　大矢　崇
東京都大田区下丸子３丁目３０番２号キヤノン株式会社内
(72)発明者　河合　智明
東京都大田区下丸子３丁目３０番２号キヤノン株式会社内

審査官　清田　健一

最終頁に続く

(54)【発明の名称】カメラ装置の制御システム、カメラシステム管理方法、及び管理装置

(57)【特許請求の範囲】

【請求項１】

複数の制御装置と管理装置とがネットワークを介して互いに接続され、前記ネットワーク上に接続されたカメラ装置を前記複数の制御装置によって制御可能とするカメラ装置の制御システムであって、

前記管理装置は、

前記カメラ装置を前記システムへ登録するための登録手段と、

前記カメラ装置への前記制御装置の接続の設定を可能とするための設定手段と、

システムに登録された前記制御装置の前記カメラ装置に対する接続状態を管理する管理手段と、を有し、

前記登録手段は、前記制御装置から発行された登録要求に応じて前記カメラ装置をシステムへ登録する設定を行い、前記設定手段は、前記制御装置から発行されたアクセス権変更要求に応じて、前記登録手段により登録された前記カメラ装置に対する、他の制御装置からのカメラ制御信号に基づく操作制御の許可及び不許可の設定変更を行うことを特徴とするカメラ装置の制御システム。

【請求項２】

請求項１において、前記複数の制御装置の前記カメラ装置に対する接続状態を表示するための表示手段を有することを特徴とするカメラ装置の制御システム。

【請求項３】

請求項２において、前記設定手段は、前記表示手段上において前記カメラ装置への接続の

設定を可能とすることを特徴とするカメラ装置の制御システム。
【請求項４】
請求項１において、前記登録手段は、前記制御装置から発行された削除要求に応じて前記カメラ装置を前記システムから削除する設定を行うことを特徴とするカメラ装置の制御システム。
【請求項５】
請求項１において、接続が許可された前記他の制御装置は、前記管理装置を介さずに前記カメラ装置へ直接接続することを特徴とするカメラ装置の制御システム。
【請求項６】
複数の制御装置と管理装置とがネットワークを介して互いに接続し、前記ネットワーク上に接続されたカメラ装置を前記複数の制御装置によって制御可能とするカメラシステムにおいて、前記カメラ装置のシステムへの登録、前記カメラ装置への前記制御装置の接続の設定、及びシステムに登録された前記制御装置の前記カメラ装置に対する接続状態の管理を行うカメラシステム管理方法であって、
前記管理装置において、前記制御装置から発行された登録要求に応じて前記カメラ装置を前記カメラシステムへ登録する設定を行い、前記制御装置から発行されたアクセス権変更要求に応じて、前記登録された前記カメラ装置に対する、他の制御装置からのカメラ制御信号に基づく操作制御の許可及び不許可の設定変更を行うことを特徴とするカメラシステム管理方法。
【請求項７】
ネットワーク上に接続されたカメラ装置を複数の制御装置によって制御可能とするシステムを管理するために、前記複数の制御装置と前記ネットワークを介して接続される管理装置であって、
前記カメラ装置をシステムへ登録するための登録手段と、
前記カメラ装置への前記制御装置の接続の設定を可能とするための設定手段と、
システムに登録された前記制御装置の前記カメラ装置に対する接続状態を管理する管理手段と、を有し、
前記登録手段は、前記制御装置から発行された登録要求に応じて前記カメラ装置をシステムへ登録する設定を行い、前記設定手段は、前記制御装置から発行されたアクセス権変更要求に応じて、前記登録手段により登録された前記カメラ装置に対する、他の制御装置からのカメラ制御信号に基づく操作制御の許可及び不許可の設定変更を行うことを特徴とする管理装置。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】
本発明は、パン、チルト等の動作が制御可能なカメラ装置の制御装置に関するものであり、特にビデオ会議システムに用いて好適なカメラ装置の制御システム、カメラシステム管理方法、及び管理装置に関するものである。
【０００２】
【従来の技術】
従来からコンピュータによってパン、チルト、ズーム等の動作が制御可能なカメラを複数台備えたビデオ会議システムが提案されているが、この提案されているシステムは、会議室等に設置された専用のシステムが主流であり、そのシステムにおいて使用される端末機器間の接続は、会議が始まると固定され、新規に端末機器を接続したり、途中で端末機器を分離したりすることは行われていなかった。
【０００３】
【発明が解決しようとする課題】
しかしながら、会議室等に設置される専用のビデオ会議システム以外に、パーソナルコンピュータ等の装置を用いた個人向けのビデオ会議システムが使用されるようになると、多数のユーザーが、必要に応じてビデオ会議やカメラ制御ソフトを実行あるいは停止する場

合が起こってくる。このような場合には、ユーザーの会議への参加あるいは離脱（システムへの端末機器の接続あるいは分離）、会議に参加しているユーザーの相互の端末機器に対してのアクセス許可の設定、およびシステムに接続されているユーザーの端末機器のアクセス状態の告知等のシステム管理を行う必要がある。

【０００４】

本発明は、上記課題を解決し、システムを停止することなく、多数のユーザーが必要に応じてネットワーク上に接続されたカメラ装置に対して制御できるようになるカメラ装置の制御システムを提供することを目的とする。

【０００５】

また、ビデオ会議への参加、離脱を容易にするカメラ装置の制御システムを提供することを目的とする。

【０００６】

【課題を解決するための手段】

上記目的を達成するため、本発明のカメラ装置の制御システムは、複数の制御装置と管理装置とがネットワークを介して互いに接続され、前記ネットワーク上に接続されたカメラ装置を前記複数の制御装置によって制御可能とするカメラ装置の制御システムであって、前記管理装置は、前記カメラ装置を前記システムへ登録するための登録手段と、前記カメラ装置への前記制御装置の接続の設定を可能とするための設定手段と、システムに登録された前記制御装置の前記カメラ装置に対する接続状態を管理する管理手段と、を有し、前記登録手段は、前記制御装置から発行された登録要求に応じて前記カメラ装置をシステムへ登録する設定を行い、前記設定手段は、前記制御装置から発行されたアクセス権変更要求に応じて、前記登録手段により登録された前記カメラ装置に対する、他の制御装置からのカメラ制御信号に基づく操作制御の許可及び不許可の設定変更を行うことを特徴とする。

【０００７】

また、本発明のカメラシステム管理方法は、複数の制御装置と管理装置とがネットワークを介して互いに接続し、前記ネットワーク上に接続されたカメラ装置を前記複数の制御装置によって制御可能とするカメラシステムにおいて、前記カメラ装置のシステムへの登録、前記カメラ装置への前記制御装置の接続の設定、及びシステムに登録された前記制御装置の前記カメラ装置に対する接続状態の管理を行うカメラシステム管理方法であって、前記管理装置において、前記制御装置から発行された登録要求に応じて前記カメラ装置を前記カメラシステムへ登録する設定を行い、前記制御装置から発行されたアクセス権変更要求に応じて、前記登録された前記カメラ装置に対する、他の制御装置からのカメラ制御信号に基づく操作制御の許可及び不許可の設定変更を行うことを特徴とする。

【０００８】

また、本発明の管理装置は、ネットワーク上に接続されたカメラ装置を複数の制御装置によって制御可能とするシステムを管理するために、前記複数の制御装置と前記ネットワークを介して接続される管理装置であって、前記カメラ装置をシステムへ登録するための登録手段と、前記カメラ装置への前記制御装置の接続の設定を可能とするための設定手段と、システムに登録された前記制御装置の前記カメラ装置に対する接続状態を管理する管理手段と、を有し、前記登録手段は、前記制御装置から発行された登録要求に応じて前記カメラ装置をシステムへ登録する設定を行い、前記設定手段は、前記制御装置から発行されたアクセス権変更要求に応じて、前記登録手段により登録された前記カメラ装置に対する、他の制御装置からのカメラ制御信号に基づく操作制御の許可及び不許可の設定変更を行うことを特徴とする。

【０００９】

【実施例】

以下、本発明の実施例を図面を用いて詳細に説明する。

【００１０】

図１は、本発明の一実施例に係わるカメラ制御装置の概略構成を示すブロック図である。

カメラ装置１０は、被写体を撮像可能でかつズーム動作を実行可能なカメラ部１２と、このカメラ部１２をパン方向（図１において左右方向）およびチルト方向（図１において上下方向）に姿勢（回動位置）制御可能な雲台部１４を有している。表示装置１６は、その画面１６ａ上にカメラ部１２によって取り込んだ画像を表示可能である。ワークステーション２０は、カメラ装置１０の状態（例えば、パン方向およびチルト方向における位置、さらにはズーム方向における位置）を制御するための命令を指示したり、カメラ装置１０によって取り込まれた画像情報を表示手段としての表示装置１６に表示させるべく、ＣＰＵ２１、不図示のＲＯＭ、ＲＡＭを有する記憶部２２、入出力部としてのＩ／Ｏポート２４，２９、ビデオボード２５を備えており、また、データを入力するためのポインティングデバイスとしてのマウス２６、およびキーボード２７が接続されている。これらの各要素はバス２８を介して接続されている。ここで、システム制御部としてのＣＰＵ２１はシステムを制御する。一方、記憶部２２のＲＯＭにはシステムを実行するためのプログラムが記憶されており、ＲＡＭには各データが記憶される。Ｉ／Ｏポート２４は、ＣＰＵ２１によって発せられるカメラの状態制御信号をカメラ装置１０に対して出力し、カメラ装置１０の各状態を示す値をカメラ装置１０から入力する。このＩ／Ｏポート２４は、本実施例においてはＲＳ２３２Ｃを介してカメラ装置１０と接続されている。ビデオボード２５はカメラ装置１０から得た画像情報を表示装置１６に送信する。本実施例においては、ＮＴＳＣ方式による信号をＲＧＢ形式に変換して出力している。

【００１１】

以上のようなカメラ制御装置２は、ネットワーク３０を介して複数接続可能であり、Ｉ／Ｏポート２９によって信号の送受信を行なう。それらの構成によって、いわゆるビデオ会議を実施可能にする。なお、本実施例においては、汎用性および処理能力を考慮して制御装置としてワークステーションを利用しているが、パソコンを利用してももちろん構わないし、専用の制御装置を用いても構わない。また、このようなビデオ会議システムにおいては、会議に聴講という形で参加も可能であるから、図１中段に示すように、カメラ装置１０が接続されていないカメラ制御装置２’をネットワーク３０に接続することも可能である。

【００１２】

図２は、本実施例を実行するためのソフトウエアの構成を示す図である。ソフトウェアの構成として以下のものがある。カメラ装置１０に接続されたワークステーション２０（厳密には記憶部２２）には、遠隔地等他の地点にカメラ装置１０によって得られた画像情報を送信するソフトとしてのビデオ会議ソフト４０、およびカメラ、制御を行うソフトとしてのカメラ制御サーバ４１が格納されている。また、このカメラ制御サーバ４１は、カメラ装置１０の起動時に後述するカメラ管理サーバ４５にカメラ装置１０がビデオ会議システム（ネットワーク）に接続されたことを登録させる。したがって、本実施例においては、カメラ管理サーバ４５を登録手段と見ることができる。

【００１３】

一方、表示装置１６に接続されたワークステーション２０（厳密には記憶部２２）には、送信されてきた画像情報の表示を行うソフトとしてのビデオ会議ソフト４２、カメラの遠隔制御を行うソフトとしてのカメラ制御パネル４３が格納されている。また、このカメラ制御パネル４３は、カメラ管理サーバ４５で管理されているリスト（詳しくは後述する）に対して参照、記述、訂正等を行うこともできる。したがって、本実施例においては、カメラ制御パネル４３は、設定手段の一部と見ることができる。また、カメラ管理サーバ４５によって、カメラ装置１０との接続が許可されると、カメラ制御パネル４３は、カメラ管理サーバ４５を介することなく、カメラ制御信号をカメラ制御サーバ４１に送信する。

【００１４】

なお、以上のソフト４１，４３をまとめて以下「カメラ制御クライアント」と称して説明する。

【００１５】

さらに、すべてのカメラ装置１０のアクセスを管理するソフトとしてカメラ管理サーバ４

５が用意される。このカメラ管理サーバ４５は、システムに接続されたいずれか一つのワークステーション２０に格納されていればよい。なお、カメラ管理サーバ４５をいずれのワークステーション２０にも格納しておき、ネットワーク上でビデオ会議システムを起動する際に、いずれか一つのワークステーション２０をサーバとして機能させてもよい。また、上記説明では、ワークステーション２０をカメラ装置１０および表示装置１６にそれぞれ接続されたものとして説明したが、通常は各ユーザーがそれぞれカメラ装置１０および表示装置１６を使用しているので、図１のブロック図に見られるように、カメラ装置１０および表示装置１６に対してワークステーション２０は１つである。したがって、ビデオ会議ソフト４０，４２は、一つのソフトとして構成される。

【００１６】

次に、本実施例の構成において、主要な動作である、カメラ制御サーバ４１によるカメラ装置１０の遠隔制御、カメラ管理サーバ４５によるユーザーの管理、カメラ管理サーバ４５によるアクセス管理について以下に説明する。

【００１７】

まず、ネットワーク３０に接続されたワークステーション２０から任意の他のワークステーション２０に接続されたカメラ装置１０を遠隔制御可能とするための実現手段について説明する。

【００１８】

カメラ装置１０は、カメラ制御サーバ４１の存在するワークステーション２０からＲＳ２３２Ｃを介してカメラ装置１０に対して制御信号を送信することにより制御が可能であるので、カメラ制御サーバ４１上に存在するパン、チルト、ズーム等のカメラ制御関数を、ＲＰＣ（Ｒｅｍｏｔｅ　Ｐｒｏｃｅｄｕｒｅ　Ｃａｌｌ　）を用いて遠隔起動することにより、この動作に対応するコマンド列を雲台部１４の不図示のＲＳ２３２Ｃポートに出力するような方法を実現すればよい。なお、この制御信号（コマンド列）は、パン、チルト等雲台１４の動作に関するものが雲台１４の不図示の制御部に送信され、ズーム等のカメラ部１２の動作に関するものがカメラ部１２の不図示の制御部に送信される。

【００１９】

そこで、カメラ制御を目的として、カメラ制御クライアントからＲＰＣを用いて遠隔地のカメラ制御サーバ４１を呼び出す関数として図３（ａ）に示すような関数を用意する。この関数の説明に当たって、図に示したコマンド等の記述は［，］でくくって記載する。以下の説明は、本実施例で採用しているＣ言語による記述に基づくが、他の言語によって記述することは構わない。図３（ａ）に示す関数において、［ｓｔｒｕｃｔ　ｃａｍｅｒａ＿ｉｎ｛～｝ｃａｍｅｒａ＿ｉｎ］　は、カメラ装置１０の動作のためにカメラ制御パネル４３からカメラ制御サーバ４１に引き渡すパラメータの組を設定するためのデータの宣言で、ここでは、チルト、パン、ズームを設定している。また、　［ｓｔｒｕｃｔ　ｆｕｎｃ＿ｏｕｔ｛～｝ｆｕｎｃ＿ｏｕｔ］は、出力値を設定するためのデータの宣言で、例えば、関数の処理結果に問題がない場合はｒｅｔ＝０，問題がある場合はｒｅｔ＝－１とすることができる。さらに、［ｃａｍｅｒａ＿ｏｐｅｎ］　は、カメラ制御サーバ４１との通信に必要なクライアントハンドラ［ＣＬＩＥＮＴ　＊ｃｌ］を作成する関数であり、［ｃａｍｅｒａ＿ｃｌｏｓｅ］は、これを消去する関数である。カメラ制御クライアントは、まず最初に、［ｃａｍｅｒａ＿ｏｐｅｎ］　を呼ぶことにより、カメラ制御サーバ４１との通信を接続した後、［ｃａｍｅｒａ＿ｐａｎ＿ｐｏｓ＿１］等の命令を実行し、最後に［ｃａｍｅｒａ＿ｃｌｏｓｅ］を呼んで、カメラの制御を終了する。

【００２０】

次に、カメラ管理サーバ４５によるユーザーの管理について説明する。

【００２１】

カメラ管理サーバ４５は、ユーザーの管理のために、図４に示すようなカメラ状態リスト５０を用いる。このカメラ状態リスト５０には、カメラ制御サーバ４１に登録されたホスト名すなわちネットワーク３０上に接続されているカメラ装置１０を示す名称、およびそのカメラ装置１０の使用状況（空き、他のユーザーがアクセス中）が格納されており、さ

らに、カメラ装置１０の位置（ｘ，ｙ，ｚ）および姿勢（パンニング角度，チルティング角度）のデータも格納されている。上記ホスト名は、本実施例で説明するように、カメラ制御装置２にカメラ装置１０が１台しか接続されていないときには、ユーザー名と一致させることができるが、複数台接続されているときには、一致させるべきではない。このリスト５０に示す例では、ｈｏｓｔ１ ，ｈｏｓｔ２ ，ｈｏｓｔ３ ，ｈｏｓｔ４ で示されるカメラ装置１０が接続されており、ｈｏｓｔ２ のカメラ装置１０をｈｏｓｔ３ のユーザーが、またｈｏｓｔ４ のカメラ装置１０をｈｏｓｔ１ のユーザーが制御可能な状態になっており、ｈｏｓｔ１ およびｈｏｓｔ３ はいずれのユーザーにも操作されていない状態を示している。

【００２２】

ここで、ユーザーの登録／削除は、カメラ制御サーバ４１の起動時に、起動したユーザー名をカメラ管理サーバ４５に告知し、カメラ状態リスト５０に登録する処理、およびカメラ制御サーバ４１の停止時に、カメラ状態リスト５０からユーザ名を削除する処理を、ＲＰＣを用いた図３（ｂ）に示す関数を使用することで実現する。図３（ｂ）に示す関数において、［ｓｔｒｕｃｔ ｈｏｓｔ＿ｎａｍｅ｛～｝ｈｏｓｔ＿ｎａｍｅ］ は、ホスト名を設定するために、カメラ制御パネル４３からカメラ制御サーバ４１に引き渡されるパラメータを設定するデータ型の宣言であり、［ＭＡＸＮＡＭＥ］ によって設定される名前の文字数が規定される。［ａｐｐｅｎｄ＿ｃａｍｅｒａ＿ｌｉｓｔ＿１］はカメラ制御サーバ４１によって起動され、カメラ管理サーバ４５内で実行される関数である。この関数が実行されると、すなわち図５（ａ）に示すように、カメラ制御サーバ４１が起動される（ステップＰ１）と、カメラ管理サーバ４５は、カメラ状態リスト５０にホスト名を登録する（ステップＰ２）。このホスト名は、図３（ｂ）からわかるように、ｈｏｓｔ＿ｎａｍｅ 型で構成され、［ｈｏｓｔ＿ｎａｍｅ－＞ｎａｍｅ］ で示される。以降、そのカメラ装置１０がカメラ制御サーバ４１を通じて制御可能であることを記憶する。また、カメラ制御サーバ４１の停止時には、図５（ｂ）に示すように、別のプログラムから［ｄｅｌｅｔｅ＿ｈｏｓｔ＿ｌｉｓｔ＿１］を起動するカメラ制御サーバ４１内の関数を呼び出し（ステップＰＰ１）、その後カメラ制御サーバ４１によりホスト名を削除してプロセスの実行を終了すればよい（ステップＰＰ２）。したがって、本実施例においては、カメラ管理サーバ４５を設定手段と見ることもできる。

【００２３】

カメラ管理サーバ４５によるアクセス管理について説明する。

【００２４】

カメラ管理サーバ４５によるアクセス管理方式として、カメラ制御クライアントによって起動され、図６に示すようなカメラ管理サーバ４５において実行される関数［ａｃｃｅｓｓ＿ｂｅｇｉｎ＿１］、 ［ａｃｃｅｓｓ＿ｅｎｄ＿１］ を追加する。この関数によってアクセスの開始と終了を実行することができる。なお、［ｓｔｒｕｃｔ ａｃｃｅｓｓ＿ｉｎ｛～｝ａｃｃｅｓｓ＿ｉｎ］ は、アクセスの対象を設定するためにカメラ制御パネル４３からカメラ管理サーバ４５に引き渡すパラメータの宣言で、［ｔａｒｇｅｔ＿ｎａｍｅ］ によってアクセスする相手を設定し、［ｕｓｅｒ＿ｎａｍｅ］ によってユーザー（自分）を設定する。

【００２５】

以下に、アクセス要求処理の手順を説明するが、その説明に先立ってまず、アクセス許可リストについて図７を用いて説明する。このアクセス許可リスト６０，６１は、カメラ管理サーバ４５が保持し、リスト６０に登録された個々のカメラ装置１０ごとに、そのカメラ装置１０にアクセス可能なユーザーのリスト６１を持っている。図７に示す例では、ｈｏｓｔ２ で示されるカメラ装置１０に対するリスト６１が示されており、ｈｏｓｔ１，ｈｏｓｔ４ で示されるユーザーがアクセス許可され得る状態にあり、ｈｏｓｔ３ のユーザーに対してはアクセスが許可されない設定となっている。なお、ｈｏｓｔ２ のユーザーは、ｈｏｓｔ２ で示されるカメラ装置１０に対しては、カメラ管理サーバ４５を介すことなく制御できるため、このリスト６１においては関与されない。

【００２６】
そして、アクセス要求処理は、図８に示すフローチャートにしたがって行なわれる。
【００２７】
カメラ制御クライアントよりカメラ装置１０に対するアクセスの要求が出されると、まず、カメラ制御クライアントがアクセスを要求するカメラ装置名を［ｔａｒｇｅｔ＿ｎａｍｅ］　より解読する（ステップＳ１）。
【００２８】
次に、カメラ状態リスト５０を参照して、そのカメラ装置１０が起動しているか否かを確認する。起動していなければ、ステップＳ７に移行する（ステップＳ２）。
【００２９】
ステップＳ２において起動していれば、［ｕｓｅｒ＿ｎａｍｅ］　を解読し、この名前で示されるユーザーがアクセスを要求するカメラ装置１０をアクセスする権利があるかどうかをアクセス許可リスト６１を参照して確認する。アクセス許可がなければ、ステップＳ７に移行する（ステップＳ３）。
【００３０】
ステップＳ３においてアクセス許可があれば、そのカメラ装置１０が他のユーザーに使用されているかどうかをカメラ状態リスト５０を参照して確認する。使用されていれば、ステップＳ７に移行する（ステップＳ４）。
【００３１】
ステップＳ４において使用されていなければ、カメラ状態リスト５０にユーザー名（ホスト名）を登録し（ステップＳ５）、アクセス要求成功（ステップＳ６）として戻る。
【００３２】
ステップＳ７においては、アクセスが失敗したと判断して戻る。
【００３３】
カメラ制御クライアント側では、アクセス操作を行う前後に必ず［ａｃｃｅｓｓ＿ｂｅｇｉｎ＿１］，［ａｃｃｅｓｓ＿ｅｎｄ＿１］を実行するようにする。
【００３４】
さらに、ユーザーごとのアクセス権の許可をユーザ自身が設定する手段を実現するために、ユーザーがカメラ管理サーバ４５内のアクセス許可リスト６０，６１を変更する命令［ｃｈａｎｇｅ＿ａｃｃｅｓｓ＿ｍｏｄｅ＿１］を設ける。これを図９を用いて説明する。
【００３５】
カメラ制御クライアントは、［ｓｔｒｕｃｔ　ｃｈａｎｇｅ＿ｉｎ｛　｝ｃｈａｎｇｅ＿ｉｎ］によって、アクセス許可を変更するホスト名を［ｔａｒｇｅｔ＿ｎａｍｅ］　に、変更するアクセス状態を［ａｃｃｅｓｓ＿ｍｏｄｅ］　に設定して、［ｃｈａｎｇｅ＿ａｃｃｅｓｓ＿ｍｏｄｅ＿１］を呼ぶ。カメラ管理サーバ４５では、まずＵＮＩＸのライブラリ関数である［ｇｅｔｈｏｓｔ］　命令を実行し、どのユーザーから命令が呼ばれたのかを確認する。次に、アクセス許可リスト６０，６１のうち、命令を呼んだユーザの項目の［ｔａｒｇｅｔ＿ｎａｍｅ］　で示されるカメラの項目を［ａｃｃｅｓｓ＿ｍｏｄｅ］　で示されるモードに変更する。ここで［ａｃｃｅｓｓ＿ｍｏｄｅ］　の内容としてアクセスを許可するモードと許可しないモードを設定する。ここで、許可するモードに設定した場合には、図７に示すリスト６１で「ＯＫ」と表示され、許可しないモードに設定した場合には「ＮＯ」と表示される。したがって、本実施例においては、カメラ管理サーバ４５を接続許可設定手段と見ることができる。
【００３６】
本実施例においては、さらに、アクセス状態をユーザーに正確に伝達し、かつ、アクセス要求やアクセス許可状態の変更等の処理を容易に行う手段を実現するために、図１０に示すようなＧＵＩ（Ｇｒａｐｈｉｃａｌ　Ｕｓｅｒ　Ｉｎｔｅｒｆａｃｅ）を画面１６ａ上に表示している。このＧＵＩにより、ユーザーは現在のカメラ装置１０の登録状況や、使用状況および使用許可状況に関する情報を得ることが可能である。また、カメラ装置１０の切り替えが可能であり、さらにアクセス許可状態の変更が可能である。以下に、図１０

に示すＧＵＩによる操作の一例について説明する。

【００３７】

ユーザーは、カメラ状態リスト更新ボタン７３を押下することにより、これまで述べてきたＲＰＣを用いる方法と同様な方法により、カメラ管理サーバ４５から現在の各カメラのアクセス情報を得ることができる。ボタンの押下は、マウス２６を操作してカーソル１８を移動させ、所望のボタン上に位置させ、第１ボタン２６ａをクリックすることにより行われる。得た情報はリストウインドウ７２に表示される。また、ユーザーは、このリストウインドウ７２上に表示されたカメラ状態リスト５０上において、マウス２６あるいはキーボード２７を操作することで制御したいカメラ装置１０の設定を行うことができる。

【００３８】

また、前述したアクセス要求処理とアクセス終了処理は、それぞれアクセス開始ボタン７４、アクセス終了ボタン７５を押下することにより実行される。また、前述したアクセス許可の変更処理は、アクセス許可設定ボタン７６を押下することにより、アクセス許可リスト６０に対応したリストウインドウ７２上に表示されるリスト上で、マウス２６あるいはキーボード２７を操作することで変更することができる。アプリケーション終了ボタン７７は、すべての操作を終了する時に押下する。したがって、本実施例においては、マウス２６あるいはキーボード２７を設定手段あるいは接続許可設定手段の一部と見ることができる。

【００３９】

また、画面１６ａ上には、画像ウインドウ７１が設けられており、カメラ装置１０により撮像した画像が表示される。ユーザーは、画面１６ａ上のカメラ制御部７８のパン操作ボタン７８ａ，７８ｂ、チルト操作ボタン７８ｃ，７８ｄ、ホームポジション復帰ボタン７８ｅ、およびズーム操作ボタン７８ｆ（拡大），７８ｇ（縮小）を操作することによってカメラ装置１０を制御することができ、この制御によって画像ウインドウ７１上に表示される画像が変わる。この制御は、マウス２６によってカーソル１８を移動させて所定のボタン上に位置させて第１ボタン２６ａをクリックすることにより行われる。なお、操作ボタン７８ａ，７８ｂ，７８ｃ，７８ｄは、パン、チルトそれぞれの移動方向に対して設けられており識別し易く、またホームポジション復帰ボタン７８ｅを設けることにより、カメラ装置１０をホームポジション（基準位置：本実施例においてはパン、チルト方向の回動角度の中央）に簡単に復帰させることができ、しかもこのボタン７８ｅを操作ボタン７８ａ，７８ｂ，７８ｃ，７８ｄが配置された中央の空きスペースに配置させたので、ホームポジションに復帰させることが視覚的にも理解し易く、ユーザーは操作し易い。なお、ズーム操作ボタンを一つにして、拡大、縮小を第１ボタン２６ａ、第２ボタン２６ｂに分けて操作するようにしてもよい。

【００４０】

以上のＧＵＩを操作してビデオ会議を行う例について説明する。

【００４１】

まず、ビデオ会議開始に先立って、ビデオ会議ソフト４０，４２、カメラ管理サーバ４５、カメラ制御サーバ４１、カメラ制御パネル４３を立ち上げる。それぞれのソフトの機能を考えると、この順番が望ましいが、カメラ装置１０の電源オンに連動させて同時に立ち上げてもよい。また、既に行われているビデオ会議に参加する場合は、カメラ管理サーバ４５を立ち上げる必要はない。

【００４２】

このような状態のもとに、ビデオ会議に参加する場合は、カメラ状態リスト更新ボタン７３を押下し、リストウインドウ７２にカメラ状態リスト５０を表示させ、このカメラ状態リスト５０中に表示されたカメラ装置１０のうちから操作したいカメラ装置１０を選択し、続いてアクセス開始ボタン７４を押下する。このとき前述した［ａｃｃｅｓｓ＿ｂｅｇｉｎ＿１］が実行される。これによって前述したアクセス処理およびカメラ装置１０の制御が可能になる。

【００４３】

ビデオ会議から抜け出す場合には、アクセス終了ボタン７５を押下する。このとき［ａｃｃｅｓｓ＿ｅｎｄ＿１］が実行される。なお、この状態では、カメラ装置１０は、ネットワーク３０上に接続されているので相手からは制御可能であり、送信されてくる画像情報も表示装置１６にてその画像を見ることができる。そして、すべての操作を終了するときに、アプリケーション終了ボタン７７を押下する。

【００４４】

なお、本実施例において、ユーザーとして説明しているものに関しては、カメラ制御装置２と見ることもできる。

【００４５】

【発明の効果】

以上説明したように、本発明によれば、システムを停止することなく、多数のユーザーが必要に応じてネットワーク上に接続されたカメラ装置に対して制御できるようになるカメラ装置の制御システムを提供することができる。

【００４６】

また、ビデオ会議への参加、離脱を容易にするカメラ装置の制御システムを提供することができる。

【図面の簡単な説明】

【図１】本発明の実施例の概略構成を示すブロック図である。

【図２】本実施例の動作を実行するためのソフトの構成を示すブロック図である。

【図３】プログラムの一例であり、（ａ）は、カメラ制御サーバ４１を呼び出す関数、（ｂ）は、ユーザーの登録、削除を行なうための関数である。

【図４】カメラ状態リストの一例を示す図である。

【図５】動作説明を行なうためのフローチャートであり、（ａ）は、カメラ起動に伴う登録処理を示し、（ｂ）は、カメラ停止に伴う抹消処理を示している。

【図６】プログラムの一例であり、アクセス処理を行なうための関数である。

【図７】アクセス許可リストの一例を示す図である。

【図８】カメラアクセス要求に対する処理を示すフローチャートである。

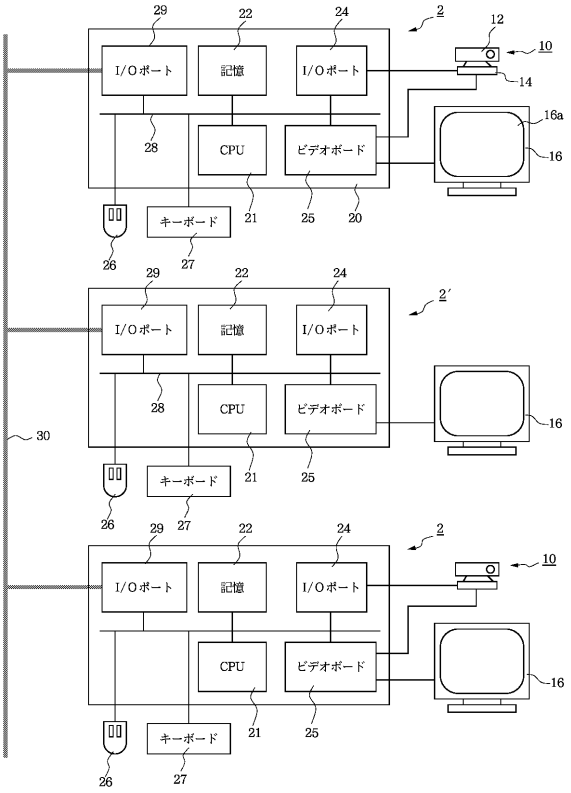
【図９】プログラムの一例であり、アクセス許可を設定するための関数である。

【図１０】ＧＵＩの一例を示す図である。

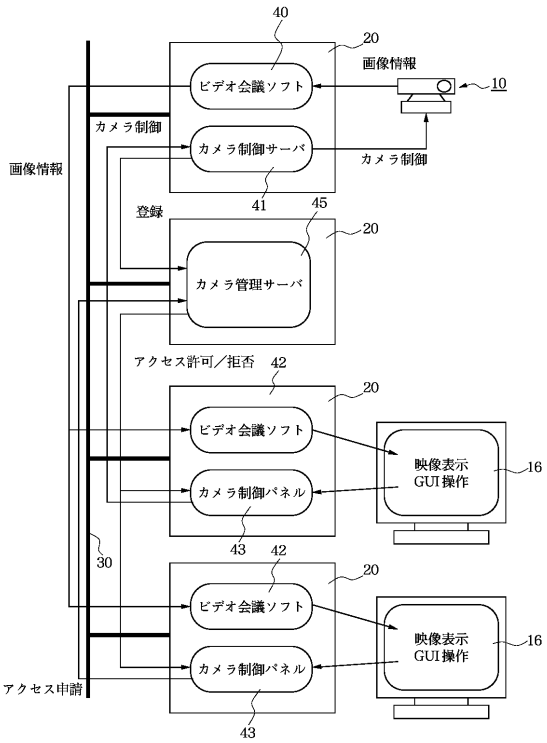
【符号の説明】

２　カメラ制御装置
１０　カメラ装置
１６　表示装置
２０　ワークステーション
２６　マウス
２７　キーボード
３０　ネットワーク
４０　ビデオ会議ソフト
４１　カメラ制御サーバ
４２　ビデオ会議ソフト
４３　カメラ制御パネル
４５　カメラ管理サーバ

【図１】



【図２】



【図３】

```
(a)
   typedef struct camera__in {
       double tilt__angle ;
       double pan__angle ;
       int    zoom ;
   } camera__in ;

   typedef struct func__out {
       int    ret ;
   } func__out ;

   CLIENT *camera__open (char   *host) ;
   void *camera__close (CLIENT *cl) ;

   func__out *camera__pan__pos__1 (camera__in *, CLIENT *cl) ;
   func__out *camera__tilt__pos__1 (camera__in *, CLIENT *cl) ;
   func__out *camera__zoom__pos__1 (camera__in *, CLIENT *cl) ;

(b)

   typedef struct host__name {
       char name〔MAXNAME〕;
   } host__name ;

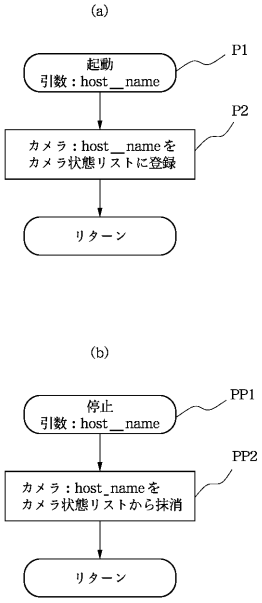
   void append__camera__list__1 (host__name *, CLIENT *cl) ;
   void delete__camera__list__1 (host__name *, CLIENT *cl) ;
```

【図４】

ホスト名	使用ホスト	位置	姿勢
host1	-------	(10, 15, 20)	(20, 35)
host2	host3	(45, 32, 20)	(0, 12)
host3	-------	⋮	⋮
host4	host1		
⋮	⋮		

50

【図５】

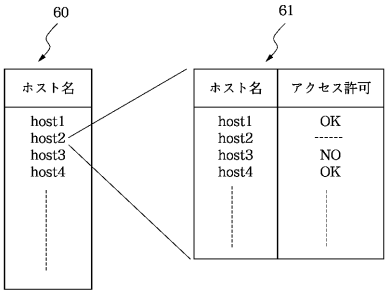


【図６】

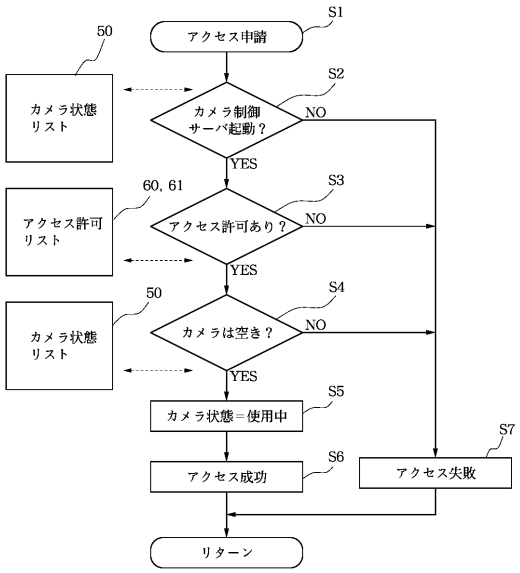
```
typedef struct access__in {
    char target__name〔MAXNAME〕;
    char user__name〔MAXNAME〕;
} access_in ;

func__out *access__begin__1 (access__in *,CLIENT *cl) ;
func__out *access__end__1 ((access__in *,CLIENT *cl) ;
```

【図７】



【図８】

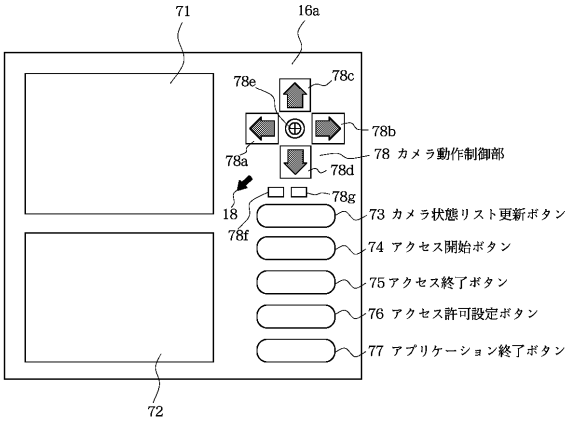


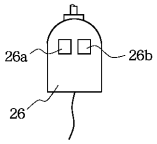
【図９】

```
typedef struct change__in {
    char tareget__name〔MAXNAME〕;
    int     access__mode;
} change__in;

func__out *change__access__mode__1 (change__in *, CLIENT *cl);
```

【図１０】





フロントページの続き

(56)参考文献 特開平０４－２９２０８６（ＪＰ，Ａ）
特開平０４－３７３２８７（ＪＰ，Ａ）
特開平０６－２３７４０８（ＪＰ，Ａ）
特開平０５－０３７９３２（ＪＰ，Ａ）
特開平０４－３１７２８８（ＪＰ，Ａ）
特開平０６－１９９２６（ＪＰ，Ａ）
特開平０６－４６４１８（ＪＰ，Ａ）
特開平０６－２６９００４（ＪＰ，Ａ）
William Gaver,Thomas Moran,Allan MacLean,Lennart LoeVstrand,Paul Dourish,Kathleen Carter,William Bux, REALIZING A VIDEO ENVIRONMENT:EUROPAC`S RAVE SYSTEM, ACM CHI`92, 米国, ACM, １９９２年 5月, pp.27-35

(58)調査した分野(Int.Cl.[7], ＤＢ名)
H04N 5/232
G06F 13/00 357
H04N 7/15 610