

平成18年度

筑波大学第三学群情報学類

卒業研究論文

題目 見られていることを気づかせる
ネットワークカメラシステムの提案と試作

主専攻 知能情報メディア主専攻

著者 樋口 潤

指導教員 田中 二郎 高橋伸 三末 和男 志築 文太郎

要 旨

近年ネットワークカメラ設置台数の増加が著しい。その背景にはネットワークカメラの設置・利用の容易さや高機能・多機能化がある。「遠隔地の人や状況の把握」に特化したネットワークカメラは、今後のユビキタスコンピューティングにも多大な貢献をもたらすと考えられる。しかしネットワークカメラを設置しようとする際、我々はプライバシー問題に関する弊害を考慮しなければならない。我々はネットワークカメラにモニタされることにしばしば抵抗感を感じるが、この抵抗感を緩和することができれば、ネットワークカメラは我々にとってさらに受け入れやすいものになるだろう。

本研究では被撮影者の「見られていること」への抵抗感を緩和するネットワークシステム：*ComeCam* を提案・試作した。*ComeCam* は被撮影者に「見られていること」を気づかせるためのフィードバックを行うことによって、「見られていること」への抵抗感の緩和を試みた。また同時に *ComeCam* は「見られていること」に気づくことから誘発されるであろうモニタする側・される側のコミュニケーションを支援する枠組みを用意し、ネットワークカメラのコミュニケーションツールとしての利用を可能とした。

目次

第1章 序論	1
1.1 研究の背景	1
1.1.1 ネットワークカメラ	1
1.1.2 利用シーン	2
1.2 ユビキタスコンピューティングとネットワークカメラ	2
1.3 「見られていること」に対する抵抗感	3
1.3.1 プライバシーの問題	3
1.3.2 「キッチン・ストーリー」に見るプライバシーの問題と今後の可能性	3
1.4 本研究の目的	4
1.5 本論文の構成	4
第2章 関連研究	5
2.1 ネットワークに接続されたカメラのプライバシー問題に配慮した研究	5
2.2 ネットワークに接続されたカメラをコミュニケーションツールとして利用した研究	5
2.3 ネットワークに接続されたカメラをコミュニケーションツールとして利用しているサービス	6
第3章 見られていることを気づかせる手法と双方向コミュニケーションの支援	7
3.1 「見られていることを気づかせる」	7
3.1.1 被撮影者に委ねられる判断	7
3.1.2 モニタ領域の必要最小限の制限	8
3.2 見られていることを気づかせるフィードバック	8
3.2.1 P.T.Z カメラの利用	8
3.2.2 フィードバック実行例	9
3.2.3 「見られていることに気づく」ことからつながる両者の関係	9
3.3 モニタする側・される側のインタラクション手法	11
3.3.1 ビデオチャットのような対話型コミュニケーション	11
3.3.2 見られている側からの能動的アクセス	12
カメラを呼び寄せる	13
モニタを拒否する	13

第 4 章	見られていることを気づかせるネットワークカメラシステム：ComeCam	15
4.1	システム概観	15
4.2	ComeCam Viewer	16
4.3	ComeCam Feedback	18
4.3.1	フィードバック手法	18
4.3.2	フィードバック判定	21
	フィードバック判定法の一例	21
	3次元空間でのユーザ・カメラ位置と視線ベクトル	23
	視線検出アルゴリズム	24
第 5 章	マーカーを用いたジェスチャによるカメラの直接的な操作	26
5.1	マーカーを用いたジェスチャ	26
5.2	マーカーの検出	27
5.3	カメラを呼び寄せる	27
5.3.1	ジェスチャと認識アルゴリズム	27
5.3.2	カメラの P.T.Z 処理	28
5.4	モニタを拒否する	30
5.4.1	ジェスチャと認識アルゴリズム	30
5.4.2	マスクの種類	32
5.4.3	マスクの削除および状態変化	32
5.5	カメラの P.T.Z に対応した 3D マスクオブジェクトの生成	34
5.5.1	ズームへの対応	34
5.5.2	パン・チルトへの対応	36
第 6 章	システムの実装	38
6.1	システム構成	38
6.1.1	ネットワークカメラ	39
6.1.2	ComeCam MainServer	39
6.2	各動作処理の解説	40
6.2.1	ログインとチャットシステム	40
6.2.2	画像の表示	41
6.2.3	カメラの P.T.Z と見られているユーザへのフィードバック	41
6.2.4	ジェスチャ認識	42
第 7 章	考察と課題	44
第 8 章	結論	46
	謝辞	47

目 次

1.1	高機能・多機能で様々な形態をもつネットワークカメラの一例	1
1.2	研究室に設置されたネットワークカメラ画像の例	2
1.3	「キッチン・ストーリー」のワンシーン	3
2.1	テレビ会議システムの例	6
2.2	ビデオチャットの例	6
3.1	P.T.Z 機能をもつネットワークカメラ	8
3.2	見られていることを気づかせるフィードバック	9
3.3	ユーザ A,B が同じ部屋にいた場合	10
3.4	お互いの関係性から誘発されるコミュニケーション	10
3.5	ビデオチャットのような対話型コミュニケーション	11
3.6	カメラを呼び寄せるジェスチャとカメラの動作処理	13
3.7	モニタを拒否するジェスチャとカメラ画像の処理	13
4.1	システムの概観	15
4.2	ComeCam Viewer	16
4.3	ComeCam Viewer 上でのビデオチャット	17
4.4	ComeCam Feedback	18
4.5	ComeCam Viewer 上での P.T.Z	19
4.6	ポップアップメッセージと音声によるフィードバック	20
4.7	フィードバック判定法の一例	22
4.8	3次元空間でのユーザ・カメラ位置	23
4.9	カメラの視線ベクトル	23
4.10	フィードバック判定	25
5.1	マーカーを用いてジェスチャをしている様子	26
5.2	ARToolKit を用いたアプリケーション例	27
5.3	マーカーを手前に引くジェスチャ	28
5.4	ジェスチャとして認識される条件を満たすマーカの動き	28
5.5	3次元空間におけるジェスチャが認識されたときのマーカー位置	29
5.6	カメラを呼び寄せるジェスチャ	29
5.7	マーカー位置を線分で結ぶ	30

5.8	マスクの位置と大きさの特定	30
5.9	モニタを拒否するジェスチャ	31
5.10	マーカーに対応した 3D マスクオブジェクト（左上から球体、立方体、クマ、クマ頭部）	33
5.11	カメラの P.T.Z への対応策の一例	34
5.12	透視投影法	35
5.13	画角が大きい場合	35
5.14	画角が小さい場合	35
5.15	ズームレベルに対応して描画される 3D マスクオブジェクト	36
5.16	パンチルトへの対応	37
5.17	パン・チルトに対応して描画される 3D マスクオブジェクト	37
6.1	<i>ComeCam</i> システム構成図	38
6.2	カメラの P.T.Z からフィードバックへの一連の流れ	41
6.3	ジェスチャ認識とその後の処理の一連の流れ	42

第1章 序論

1.1 研究の背景

近年では技術の進歩により、カメラを IP ベースのコンピュータネットワークに接続することが可能となり、Web サイトを通じて全世界の人々がライブ映像を見ることができるようになった。現在でもその用途は多様であるが、今後のユビキタスコンピューティング環境の整備・発達を考慮しても、我々の周りに様々な形態のネットワークカメラがあふれる時代がやってくるのが予想される。

1.1.1 ネットワークカメラ

ネットワークカメラは、カメラとコンピュータが一体化したものと考えることができ、個々に IP アドレスを持つ。ネットワークに直接接続され、PC につなげる必要がなく独立して動くので、IP ネットワークさえあればどこにでも設置することができる。これに対し Web カメラは、動かすために USB や IEEE1394 ポートを使って PC につなげる必要がある。したがってネットワークカメラは設置や利用が非常に容易であり、利用に伴い必要となる費用も廉価である [1]。

近年では単に画像を送信するだけでなく、様々な機能を持つネットワークカメラが開発されている（図 1.1¹）。P.T.Z（パン・チルト・ズーム）のようなカメラコントロール、動作検出、音声通信、無線 LAN、携帯電話からの画像閲覧に対応、といった高機能・多機能化により、ネットワークカメラは今や多様なニーズに対応できるメディアツールとなった。



図 1.1: 高機能・多機能で様々な形態をもつネットワークカメラの一例

¹左から

AXIS: AXIS 212 PTZ <http://www.axiscom.co.jp/prod/212/>

コレガ: CG-WLNCPTG <http://corega.jp/product/list/others/wlncptg.htm>

Mega Chips: Surfeel m101i <http://www.mcsc.jp/products/surfeel/>

1.1.2 利用シーン

ネットワークカメラは防犯・セキュリティ対策といったいわゆる監視カメラとしての利用以外に、遠隔地からの子供やペットのモニタリング、観光地などのライブ映像の配信、店舗の陳列商品の売れ行き・在庫把握などで利用されている。また図 1.2 のように、遠隔地に分散されたオフィスや研究室にネットワークカメラを設置する場合がある。この主な目的としてはプレゼンス情報の取得にある。ネットワークカメラは設置場所の状況をそのままリアルタイムな映像として提供するので、在室状況の他、今何をしているのかといった周辺情報の取得にも役立つ。

このように多様な用途をもつネットワークカメラだが、その利用の主な目的としては「遠隔地の様子をモニタする」ことに集約される。加えて設置・利用の容易さや高機能・多機能化が、その利用数の増大に拍車をかけていると言える。



図 1.2: 研究室に設置されたネットワークカメラ画像の例

1.2 ユビキタスコンピューティングとネットワークカメラ

ユビキタスコンピューティングとは、アメリカのマークワイザーによって 1991 年に提唱された現実世界にコンピュータが偏在する概念である。ここでは特にユーザを取り巻く環境や状況を計算機が把握し、ユーザのタスクの処理に最適で、かつ環境や状況に応じたサービスや情報を提供する技術・環境のことを指す。その際、ユーザを取り巻くコンテキスト情報をいかに取得するか、つまり「人の行動をどう把握するか」が重要となってくる。

この「人の行動をどう把握するか」という問題について、ネットワークカメラの利用を考えることができる。ネットワークカメラはこの遠隔地のモニタリングを本来の用途とすると

ころであり、設置・利用の容易さや高機能・多機能化を考慮した上で、ユビキタスコンピューティングに対し大きな貢献をもたらすと考えられる。しかしネットワークカメラの設置を考える際、我々はプライバシーに関する弊害を考慮しなければならない。

1.3 「見られていること」に対する抵抗感

1.3.1 プライバシーの問題

我々はネットワークカメラにモニタされることに、しばしば抵抗感を感じる。それは自身のプライバシーが脅かされる不安によるものである。ネットワークカメラを利用する際、プライバシーの問題は避けて通ることができない。例えば図 1.2 のような状況においては、カメラのモニタする側とされる側は既知の間柄であるはずであり、モニタされる側は撮影されることにある程度同意していると考えられる。しかしこの場合ネットワークカメラは常時稼動しており、ディスプレイやデスクといった作業領域や日常の生活行動など、いくら既知の間柄といえど保持すべきプライバシーがときに存在するはずである。ネットワークカメラの設置運用側と被撮影者側の利益は相反することが多いが、現在は設置数の増大とは裏腹に、設置運用者側の利益に重みを増している傾向が強い。この現状を見ながら大田 [2] はモニタリング用のカメラを指して「我々の好むと好まざるとに関わらず、ますます増大していくもの」と述べている。

1.3.2 「キッチン・ストーリー」に見るプライバシーの問題と今後の可能性

「キッチン・ストーリー」[3] は 1950 年代の北欧を舞台にした映画である。ノルウェーの田舎に住む年老いた男性の元に、一人の調査員がやってくる。調査員の目的は「独身男性の台所における動線調査」であり、調査員は図 1.3 のように男の家の台所に監視台を設置し、男を観察し始める。男は当初見られることに反発し、様々な抵抗を試みる。調査には「調査対象と話してはならない」などの厳しいルールが設けられていたが、2 人はある日そのルールを破り会話をしてしまい、その後 2 人は次第に交流を深めていく。



(C)2003 BULBUL FILM AS

図 1.3: 「キッチン・ストーリー」のワンシーン

我々はこの映画に「行動把握を目的とする撮影者」と「見られることへ抵抗を感じる被撮影者」の明示的な関係性を見ると共に、その状況を打破することにより生まれる「両者の交流」という可能性を見出すことができないだろうか。

被撮影者のネットワークカメラに対する「見られていること」への抵抗感を緩和することができれば、ネットワークカメラを今まで以上に受け入れやすいものとすることができ、ユビキタスコンピューティングにおけるさらなる発展を望むことができる。また同時に我々はネットワークカメラ本来の利用用途を拡張することもできると考える。

1.4 本研究の目的

本研究は被撮影者の「見られていること」への抵抗感を緩和するネットワークカメラシステムを提案・試作する。

我々は「見られていること」への抵抗感を緩和するための手法として、カメラにモニタされていることを気づかせるフィードバック手法を提案する。またモニタされていることを気づかせることによって、モニタする側・される側のつながりが改めて意識されることが想定できる。我々はそうした両者の関係から誘発されるであろうコミュニケーションを支援する。

1.5 本論文の構成

本論文の構成は以下のとおりである。

第2章では本研究と関連する研究を述べる。第3章では本研究の提案手法を提示し、第4章・5章でそれらの手法を具体的に実装したシステムのアプリケーションについて解説する。第6章ではシステム全体の構成や各動作処理、および実装について解説する。第6章でシステムの考察を行い今後の方向性を述べ、第7章で本論文をまとめる。

第2章 関連研究

本章では本研究と関連する先行研究、ならびに現行のサービスについて述べる。

2.1 ネットワークに接続されたカメラのプライバシー問題に配慮した研究

ネットワークベースのカメラに付随するプライバシー問題に考慮した研究は数多くある。[4, 5, 6] は遠隔地に分散された作業空間にネットワークカメラが設置された場合に、協調作業におけるアウェアネスとプライバシーのトレードオフについて指摘しながら、そのバランスについて議論している。特に [5] は数段階のピクセルレベルのブラーフィルターと生成されるアウェアネスの関連を述べた。[6] は在宅勤務者の作業空間に焦点をあて、来訪者を検知する仕組みなどを導入した。またカメラの撮影状態をユーザに視覚的、または音声でフィードバックを行う手法を実装している。

2.2 ネットワークに接続されたカメラをコミュニケーションツールとして利用した研究

本研究はネットワークカメラの利用用途の拡張として、コミュニケーションを支援する枠組みを考案するが、同様にネットワークベースのカメラをコミュニケーションツールとして利用した先行研究がある。同じコミュニケーションツールとしての枠組みでありながら、そのアプローチは多岐に渡る。[7] はカメラの画像をプレゼンス情報として利用するシステムであり、画像だけでは伝わりにくい情報を漫画の効果線などを重ね合わせて表現している。またカメラ画像の双方向通信により遠隔地間の同室感を実現する研究があるが、[8, 9] は視線や指差しといったノンバーバルな情報を共有することで、円滑な会話や共同作業空間の形成を支援している。[10] は連続したカメラ画像をアルファブレンディングで重ね合わせることで、プライバシーに配慮しつつ離れた空間で多様なアウェアネスを生むためのシステムを提案した。

2.3 ネットワークに接続されたカメラをコミュニケーションツールとして利用しているサービス

ネットワークベースのカメラをコミュニケーションツールとして利用している現行のサービスの一つに、テレビ会議システムがある（図 2.1¹）。これは遠隔地にある会議室やユーザをネットワークカメラベースで接続したものである。テレビ会議システムは、人々が同じ場所に物理的に集まる手間を排除し、お互いの表情や身振り手振りを見ることができる点で対面での会議と同じ効果をもたらす。

同様のサービスにビデオチャットがある（図 2.1²）。ビデオチャットはテレビ電話のように音声や動画でリアルタイムに会話するシステムだが、ビデオチャットは文字ベースのチャットに音声や動画を追加したもので、音声ベースのテレビ電話とは適用場面が異なる。ビデオチャット機能は多くのインスタントメッセージに採用されている。



図 2.1: テレビ会議システムの例

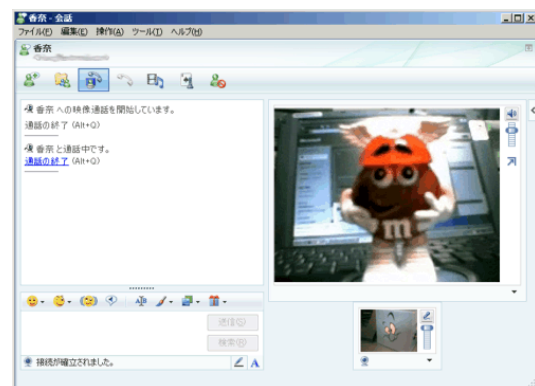


図 2.2: ビデオチャットの例

¹NEC コミュニケーションドアエクスプレス <http://www.nec.co.jp/middle/commdoor/products/express/>

²Windows Live Messenger <http://messenger.live.jp/>

第3章 見られていることを気づかせる手法と双方向コミュニケーションの支援

本章では本研究の提案手法について述べる。

まず「見られていること」への抵抗感を緩和するための手法として、被撮影者に「見られていることを気づかせる」手法を提案し、解説する。次に被撮影者に「見られていることを気づかせる」ための具体的な手段として、フィードバック手法を紹介する。最後に見られていることに気づくことから誘発されるであろう双方向コミュニケーションを支援するための枠組みを解説する。

3.1 「見られていることを気づかせる」

例えば街頭や店内の監視カメラの存在に気がついたとき、そのカメラの存在を強く意識したり、そのカメラの視野内に入らないように行動する状況はよくあることである。このようにカメラに対して「見られていることに気づいている」状況下にあるならば、被撮影者は容易にその対応策をとることができる。つまりカメラに対して見られているかどうかの判断を行うことができれば、被撮影者のカメラに対する抵抗感はいくらか緩和されるだろう。本研究では被撮影者のネットワークカメラに対する「見られていること」への抵抗感を緩和するための手法として、被撮影者に「見られていることを気づかせる」手法を提案する。

3.1.1 被撮影者に委ねられる判断

被撮影者側が撮影に同意しているような状況では、被撮影者側の不利益は生じず、プライバシーの問題は生じない。「見られていることを気づかせる」という手法は、被撮影者に対して撮影の了承をとることに等しいと考えられる。つまり見られていることに気づいた後、撮影に了承しない（すなわち撮影に対して対応策をとる）、撮影に了承する（すなわちそのまま撮影を続行させる）という判断は、被撮影者に委ねられることになる。またその際、特に「〇〇に見られている」という情報まで被撮影者が取得できるとすると、その後の判断は「〇〇になら見られてもよい」、「〇〇に見られるのは好ましくない」といったより具体的なものになるだろう。

3.1.2 モニタ領域の必要最小限の制限

またネットワークカメラを設置する場合、遠隔地のモニタリングという特性からそのカメラは常時稼動している可能性が高い。先に述べたように図 1.1 のようなネットワークカメラ設置環境では、被撮影者はそうしたカメラの稼動形態にある程度同意している。しかし時として保持したいプライバシーが被撮影者にも存在し、それは突発的に発生するものと考えられる。したがって「いつ守ればよいかわからない」被撮影者のプライバシーを保持するために、視野内の大部分の画像をモザイクなどで加工したり、被撮影者が常に（あるいは長時間）カメラの視野に入らないような工夫をしてしまうと、「遠隔地の様子をモニタする」というネットワークカメラ本来の役割を果たすことができなくなってしまう。

我々の提案手法を用いた場合、カメラにモニタされていないときはプライバシーに関して考慮する必要は当然ないし、カメラにモニタされているとき、被撮影者は見られていることに気づいているので、モニタ領域の制限などの対応は被撮影者の判断にて実行される。つまり被撮影者が「見られていることに気づく」ということは、常時稼動するネットワークカメラのモニタ領域に対し、その制限を必要最小限に抑えることができる。

3.2 見られていることを気づかせるフィードバック

ネットワークカメラに対して見られていることを気づかせることを、本研究では被撮影者にフィードバックを与えることで実現する。以降はその具体的な方法について説明する。

3.2.1 P.T.Z カメラの利用

本研究においては遠隔地のモニタリングのためにネットワークカメラの利用を想定するが、特に図 3.1 のような P.T.Z（パン・チルト・ズーム）機能をもつものを利用することを考える。その理由として、1 台で部屋全体を網羅できる幅広い視野や高精度なズーム倍率を活用することはもちろんであるが、後述する「人の視線のメタファ」として利用するためでもある。



図 3.1: P.T.Z 機能をもつネットワークカメラ

3.2.2 フィードバック実行例

提案手法による、見られていることを気づかせるためのフィードバック実行例を図 3.2 に示す。遠隔地に分散した部屋があり、そこにあるカメラ設置運用者と被撮影者が存在する。便宜上、ここでは彼らをユーザ A、ユーザ B と呼ぶ。今ユーザ A が、ユーザ B のいる部屋をモニタしているとする。そして P.T.Z. によってユーザ A がユーザ B を視野中心付近にもってきて注視する。このときユーザ B に「今ユーザ A に見られているよ!」というフィードバックを与える。ユーザ B はフィードバックによって「今自分はユーザ A によってモニタされている」という状況に気づくことができる。

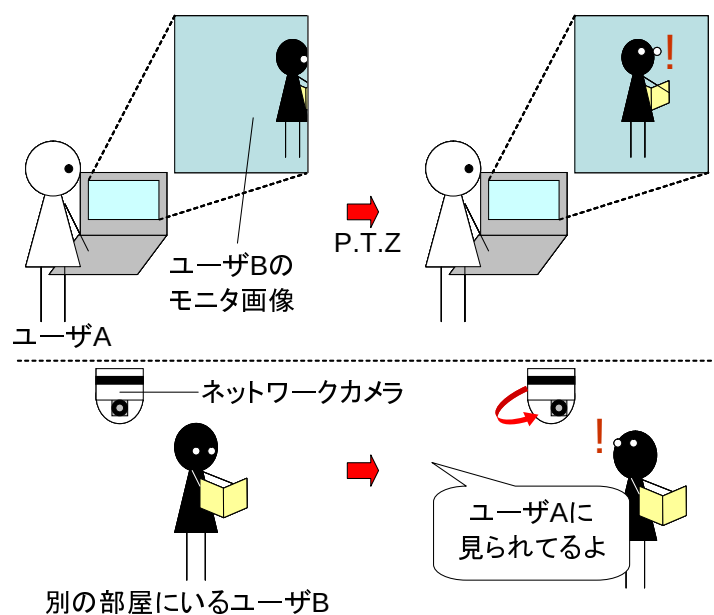


図 3.2: 見られていることを気づかせるフィードバック

3.2.3 「見られていることに気づく」ことからつながる両者の関係

図 3.2 に示した実行例は、同じ部屋にいて「ユーザ A がユーザ B を見て、ユーザ B がそれに気づいた」という状況（図 3.3）に等しく、遠隔地にあるユーザに同室感を与える。つまり、ここではネットワークカメラは「ユーザ A の視線のメタファ」としての役割を果たしていると言える。このフィードバック手法によって、同じ部屋なら当然気づくであろう「見る・見られている」という関係を、遠隔地におけるユーザ間で明確にすることができる。

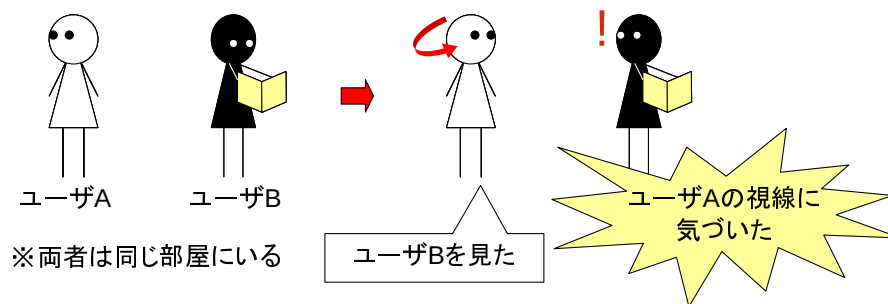


図 3.3: ユーザ A,B が同じ部屋にいた場合

「見る・見られている」という関係が明確となるということはお互いの関係に気づく、あるいはお互いの関係を意識するということであり、両者にその後コミュニケーションを誘発させる効果が期待できる。例えば図 3.3 のように同じ部屋において「ユーザ A がユーザ B を見て、ユーザ B がそれに気づいた」という状況が発生したとするならば、その後図 3.4 に示すように、両者の間に会話のようなコミュニケーションが発生することは、容易に想像できるだろう。

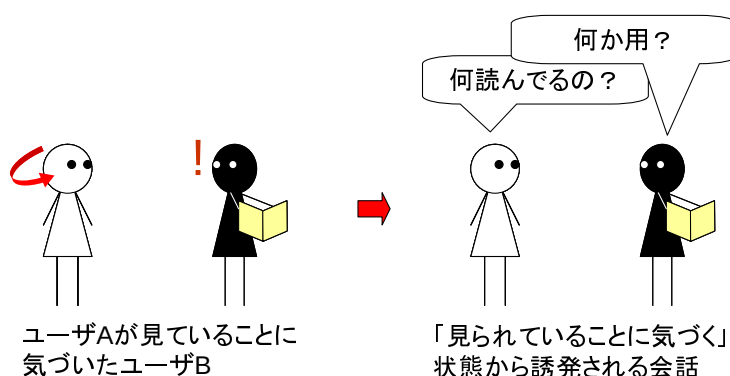


図 3.4: お互いの関係性から誘発されるコミュニケーション

「見られていることを気づかせる」ということは、単にプライバシーを保護することに留まらない。例えば「〇〇に見られている」という情報を提示された被撮影者は、カメラを介して自分をモニタしている撮影者のことを意識するだろう。逆にモニタする側も、自分の視線に気づいた被撮影者を改めて意識するだろう。このように両者は自分たちのつながりを改めて認識することができ、そうした両者の間に何らかの方法でコミュニケーションを図りたいという要求が生じるのは自然な流れであると言える。

3.3 モニタする側・される側のインタラクション手法

見られていることに気づくということは、すなわち両者がお互いの関係を意識している状態である。さらに両者のインタラクション手法を用意することにより、ネットワークカメラを介した遠隔地における同期型双方向コミュニケーションが達成されることが考えられる。

3.3.1 ビデオチャットのような対話型コミュニケーション

両者のインタラクション手法としてまずチャット、音声通信などの同期的かつ直接的な対話と考えられる。特にネットワークカメラの画像を用いることにより、相手や周囲の様子を見ながら対話することができ、直接対話するような自然で臨場感のあるコミュニケーションを行うことができるだろう。

本研究では図 3.5 のように自分のいる部屋も含め、複数の部屋の様子をモニタすることができるビューアを用意し、そのビューアで同時にチャットも行うことができるようなインタフェースを提案する。同じウィンドウに複数のカメラ画像が表示されるので、お互いの顔をズームすれば遠隔地間のビデオチャットのような使い方もできる。通常ビデオチャットを利用する際は各人の PC に Web カメラを取り付ける必要があるが、P.T.Z 機能をもつネットワークカメラを用いることにより、1 台で部屋中のユーザをほぼカバーすることも可能である。またズームレベルを調整すれば個人に限らず複数人を撮影することもでき、テレビ会議システムの延長としての利用も可能であろう。

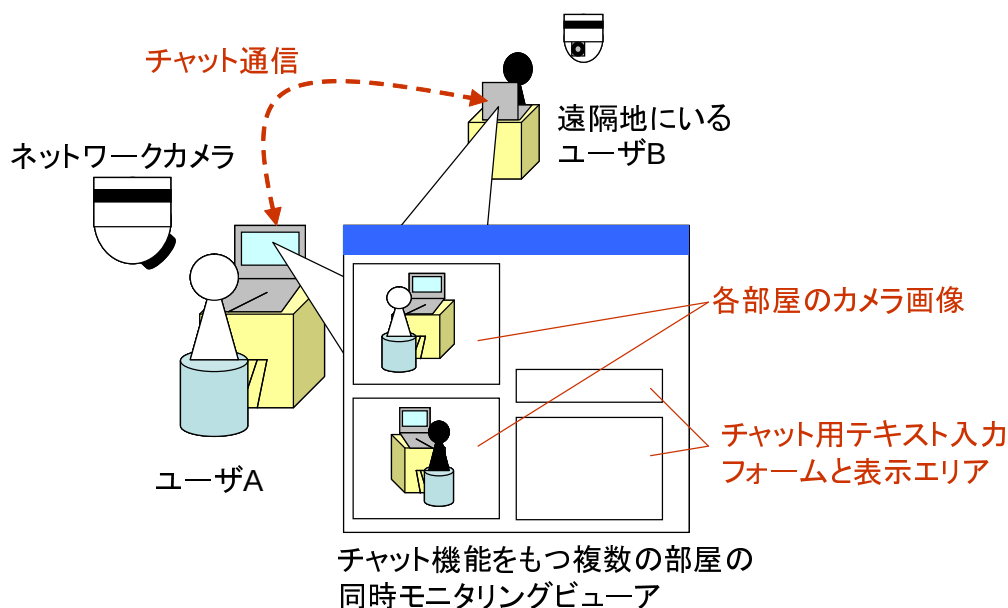


図 3.5: ビデオチャットのような対話型コミュニケーション

このチャットシステムの利用シーンを考える。モニタされていることに気づいたユーザが

ビューアを起動する。そして自分がいまどのように撮影されているかを確認するとともに、遠隔地にいる自分をモニタしているユーザを確認する。そしてチャットでお互いに対話を始める。つまりこの状況においては「モニタする側もまた、モニタされる側」であり、撮影者側の一方的なモニタリングであった従来のシステムに比べ、両者の関係は対等に近いものになったと言える。

3.3.2 見られている側からの能動的アクセス

これまで本章では「見られていること」への抵抗感を緩和するため、被撮影者に見られていることをフィードバックとして提示する手法を提案し、それから誘発されるであろうコミュニケーションを支援するため、ビデオチャットのような対話型コミュニケーション手法を提案した。これにより被撮影者のプライバシーに対する不安感はいくらか払拭された。また一方的であったモニタリング体制は解消され、双方向的なコミュニケーションが行われることになった。しかしながらこれらの一連の手法はまだカメラ撮影者からのアクセス始動の動作処理であり、被撮影者からもカメラにアクセスすることを可能とすることができれば、カメラを仲介したインタラクションはより深めることができるだろう。

また大田 [2] はモニタリング用カメラが今後も確実に増大するものであると述べ、その前提の上で我々がそれと付き合うためには、「安全」というプライバシーを犠牲にして得られるメリットの他に、さらなる付加価値を考慮する必要があると述べている。それと同時に、新しい付加価値の在り方として、従来カメラ設置者のみが利用していたカメラ（映像）を被撮影者からも利用可能とすることが一つの柱であるとしている。我々は新しい付加価値の在り方としてネットワークカメラをコミュニケーションツールとして利用する枠組みを考えるが、そのためにはカメラ（映像）を被撮影者からも利用可能とし、さらに前述のように被撮影者からもインタラクティブにアクセスを可能とする必要がある。

本研究では被撮影者がカメラに対して能動的にアクセスし、カメラあるいはカメラ画像を直接的に操作する手法を提案する。結果として被撮影者とカメラとのインタラクティブ性をサポートし、被撮影者はいつでも必要なときにカメラ（画像）にアクセスできる。アクセス手法としてはカメラに対する直接的なジェスチャを用いる。なおここでのジェスチャとは、実行したい操作をカメラに対して身振り手振りで表現することを指す。ジェスチャを採用する理由として、被撮影者は（モニタしている側とは違い）常にコンピュータを操作している状況とは限らないことが挙げられる。PC等を介さずカメラに対して直接ジェスチャするという手法を用いることで、カメラとのインタラクションを実現し、そうした問題に対処できるのである。本研究の提案手法では、ジェスチャにより以下の2つの動作処理を実行可能とする。

カメラを呼び寄せる

図 3.6 に本動作処理例を示す。被撮影者がモニタをしている人間の興味を引くために、カメラを自分あるいは特定の場所に対して P.T.Z させたいと思ったとする。その際、被撮影者はカメラに対して「カメラこっち来て!」というジェスチャをする。するとカメラがジェスチャを行った領域に対して P.T.Z 処理を行う。

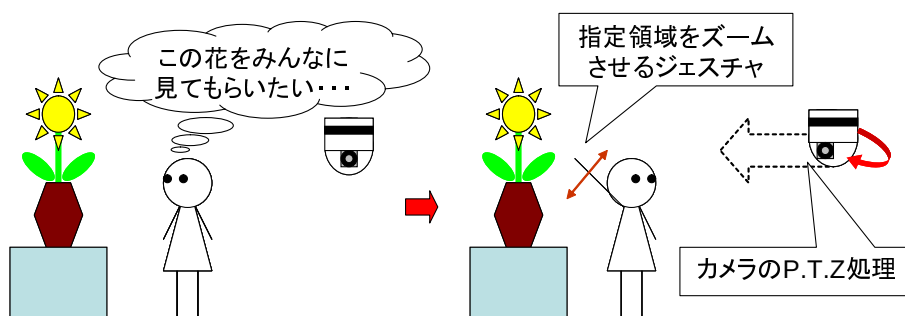


図 3.6: カメラを呼び寄せるジェスチャとカメラの動作処理

モニタを拒否する

図 3.7 に本動作処理例を示す。まずモニタされていることに気づいた被撮影者が、自分が現在撮影してほしくない状況下にあると思ったとする。その際、被撮影者はカメラに対して「写さないで!」というジェスチャをする。するとカメラ画像のジェスチャで指定した領域にマスクがかかり、被撮影者のプライバシーが保護される。これは先に述べた見られていることに気づいた被撮影者が、撮影に対して行う対応策の一つと考えられる。

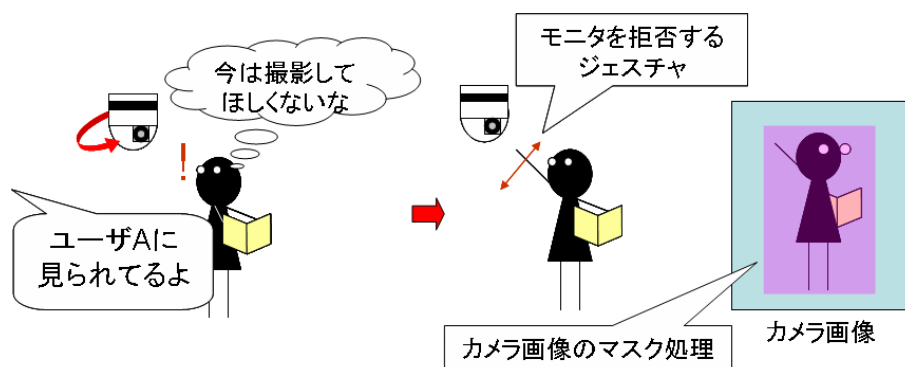


図 3.7: モニタを拒否するジェスチャとカメラ画像の処理

先の同室感の観点から述べるのであれば、前者は「向こうにいる人を『こっち来て！』と呼ぶ」、後者は「『見ないで！』とこちらを見ることを拒否する」という動作とそれぞれ等しいと言える。つまりジェスチャという人間にとって直感的なアクセス手法を、カメラに対して直接用いることで、カメラの向こう側（すなわち遠隔地）にいるユーザに身振り手振りで直接的にコンタクトするのと等しい感覚が得られる。人間にとって直感的なインタフェースをネットワークカメラに用いた研究としては、手振り検出を用いた [11] や携帯端末を用いた [12] などが挙げられるが、カメラと人間の自然なコミュニケーションを実現するためには、人間にとって自然なアクセス手法が望ましいことが言える。本研究ではジェスチャという直感的なアクセス手法により、モニタする側・される側のインタラクションはより深めることができる考える。

第4章 見られていることを気づかせるネットワークカメラシステム：ComeCam

我々は被撮影者に見られていることを気づかせるネットワークカメラシステム：ComeCamを試作した。ComeCamは第3章で提案した「見られていることを気づかせるフィードバック」を具体的実現するためのアプリケーションを有する。

本章では ComeCam の概観および各アプリケーションについて解説する。

4.1 システム概観

本システムの概観を図 4.1 に示す。本システムにおいては遠隔地にある部屋それぞれに先に述べた図 3.1 のような P.T.Z 機能をもつネットワークカメラを設置する。そしてカメラの被撮影者でありシステム利用者でもあるユーザの PC では、ComeCam Viewer と ComeCam Feedback と呼ばれるアプリケーションが動作する。

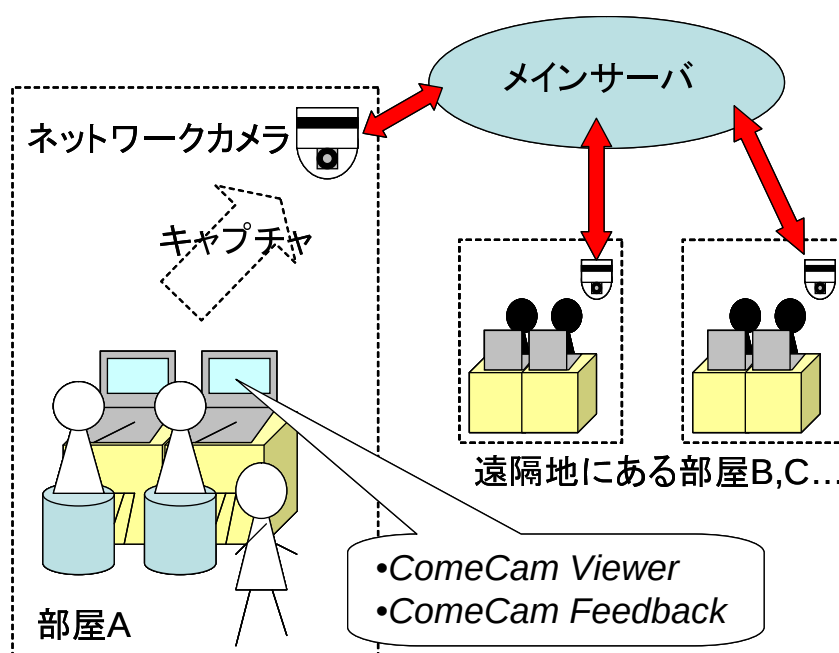


図 4.1: システムの概観

4.2 ComeCam Viewer

ComeCam Viewerはネットワークを介し取得された複数のカメラのキャプチャ画像を、ディスプレイ上で同時モニタするための基本的なビューアアプリケーションである。図4.1はComeCam Viewer プロトタイプのスナップショットであり、遠隔地にある2つの研究室の画像を表示している。



図 4.2: ComeCam Viewer

- ① カメラのキャプチャ画像：遠隔地にある部屋それぞれに設置されたネットワークカメラによるキャプチャ画像である。画像をクリックすることにより、クリックした場所をカメラ視野の中心位置となるようにパン・チルトすることができる。またマウスホイールを動かすことにより、現在マウスカーソルがある画像のズームイン・ズームアウトを実行する。
- ② ユーザ名入力フォームと Login ボタン：ComeCam Viewer は起動した直後には画像は表示されず、後述するチャット機能も使用できない。起動後はこのフォームにユーザ名を入力し、Login ボタンをクリックすることによってシステムにログインできる。プライベート

シー保護の観点から本システムは利用者を特定する必要があり、ユーザ認証の仕組みを有する。ログインが完了するとウィンドウ上に画像が表示され、チャット機能も利用可能となる。

- ③ quit ボタン：このボタンをクリックするとシステムからのログアウトが実行され、画像更新が停止しチャット機能が利用不可能になる。
- ④ テキスト入力フォームと送信ボタン：ComeCam Viewer はチャット機能を有しており、ここにテキストを入力して送信ボタンをクリックすることにより、システムにログインしている全てのユーザにテキストを送信することができる。
- ⑤ テキスト表示エリア： の入力フォームで入力したテキストを表示するディスプレイである。テキストの前に送信者のユーザ名が冠される。

④や⑤の説明にあるとおり、ComeCam Viewer では付加的な機能としてログインユーザ間でのチャット通信ができる。これにより図 3.5 で示したようなビデオチャットのような対話型コミュニケーションが可能となる。図 4.3 は実際に ComeCam Viewer 上でお互いの周辺画像をズームし、相手の顔画像を伴いながらチャット通信を行っている様子を示す。

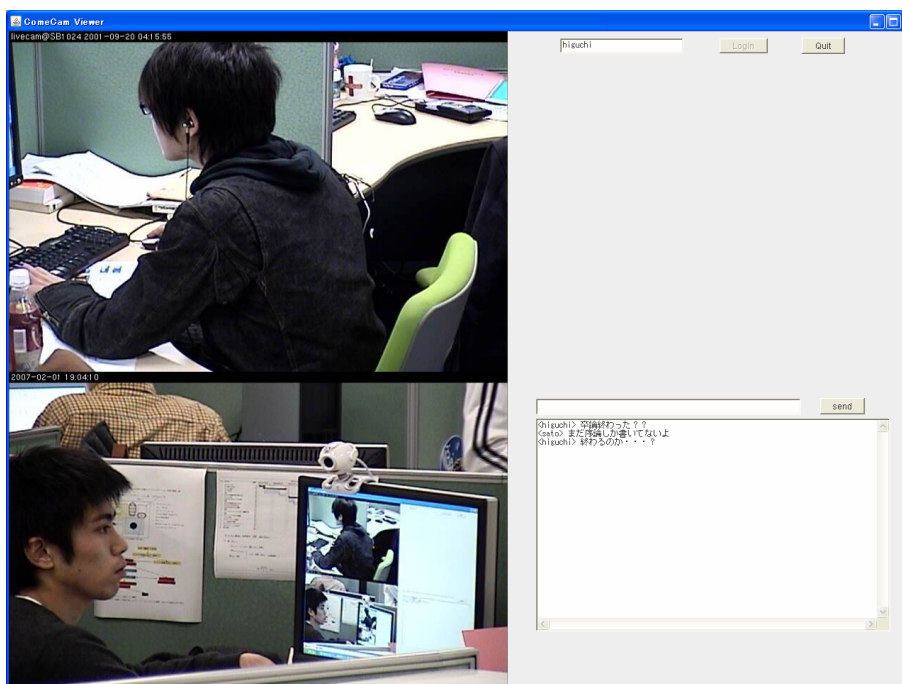


図 4.3: ComeCam Viewer 上でのビデオチャット

4.3 ComeCam Feedback

ComeCam Feedbackはカメラにモニタされていることをユーザに気づかせる常駐アプリケーションである。アプリケーションを起動するとシステムトレイに図4.4のようなアイコンが表示され、スタンバイ状態となる。

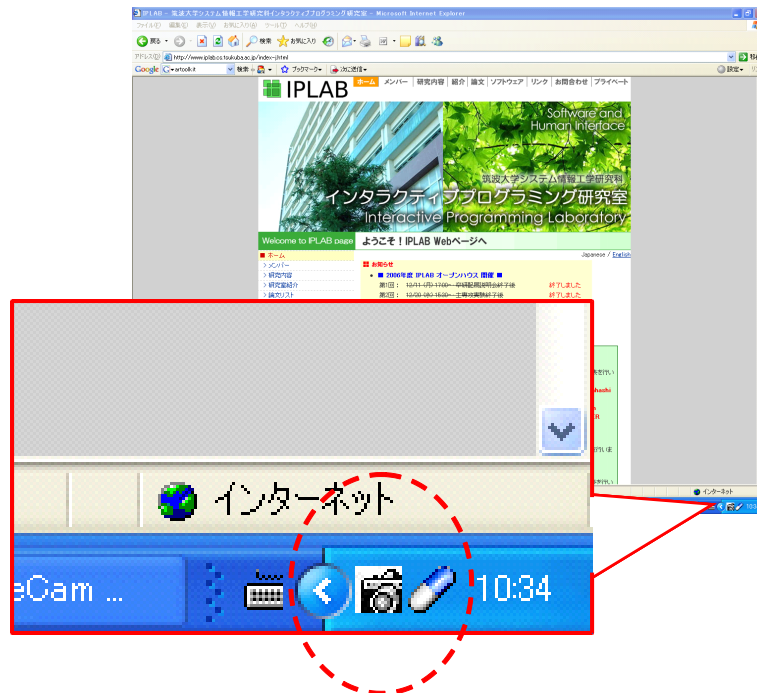


図 4.4: ComeCam Feedback

4.3.1 フィードバック手法

ComeCam Viewer 上にて図4.5のようにカメラ画像をクリックすることによってP.T.Zを行った際（図4.5では上段のカメラ画像においてP.T.Zが行われている）、カメラ視野内のユーザがフィードバック対象になるかどうかの判定を行う（後述、4.3.2）。そして図4.5のように視野内にフィードバック対象となるユーザが存在すると判定された場合、対象ユーザのPCにてスタンバイ中のComeCam Feedbackが「〇〇に見られている」というフィードバックを行う。

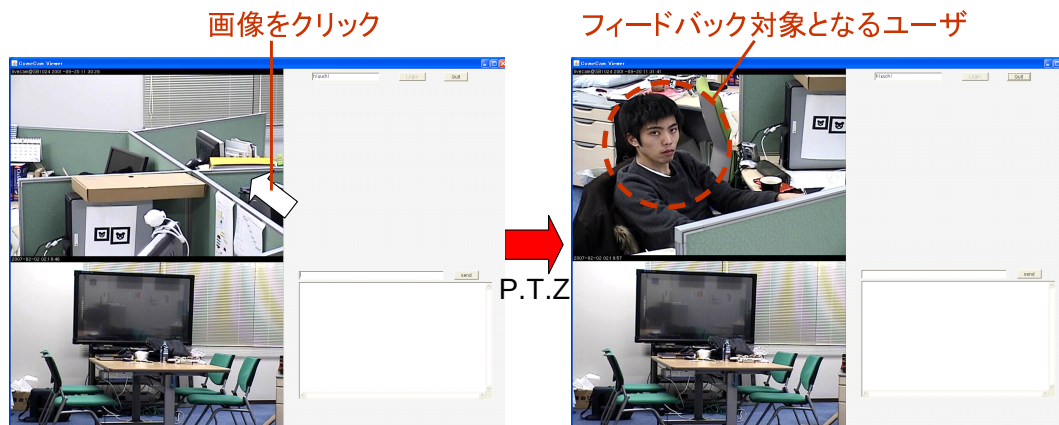


図 4.5: ComeCam Viewer 上での P.T.Z

フィードバック手法として図 4.6 のようなポップアップメッセージが表示される。メッセージの内容は「対象ユーザをモニタしているユーザ名」、「カメラのズームレベル（100 を最高値とする）」という情報を提示する。これらの情報はフィードバックをされたユーザの今後の対応（撮影を了承するか否か、自分をモニタしているユーザとコミュニケーションを図るか否か）を具体的にするための支援を行う。例えばフィードバックによって「見られていることに気づいた」状態であっても、メッセージの内容によって以下のような対応の変化が生じると考えられる。

- 『ユーザ A に見られています（ズームレベル：20）』：「ユーザ A になら見られてもかまわない」と撮影を続行させる。
- 『ユーザ B に見られています（ズームレベル：20）』：「ユーザ B に見られるのは好ましくない」とモニタ拒否の対応を検討する。
- 『ユーザ A に見られています（ズームレベル：80）』：「ユーザ A になら見られてもかまわないが、こんなに自分をズームしてモニタするのは、何か用でもあるのか」と ComeCam Viewer のチャット通信にて、ユーザ A にコンタクトをとろうとする。

またメッセージの表示と同時に独自の効果音が鳴るようになっていて、ディスプレイを注視していなくても「見られている」という自身の状態に気づくことが可能である。なお効果音もズームレベルによって違う音が鳴るようになっており、ズームレベルが高いほど注意を喚起するような効果音が選択される（効果音の選択はシステム設計者により一意的に行われた）。したがって音声によるフィードバックもまたメッセージによるそれと同様に、ユーザの今後の対応のための判断材料の一つになると考えられる。

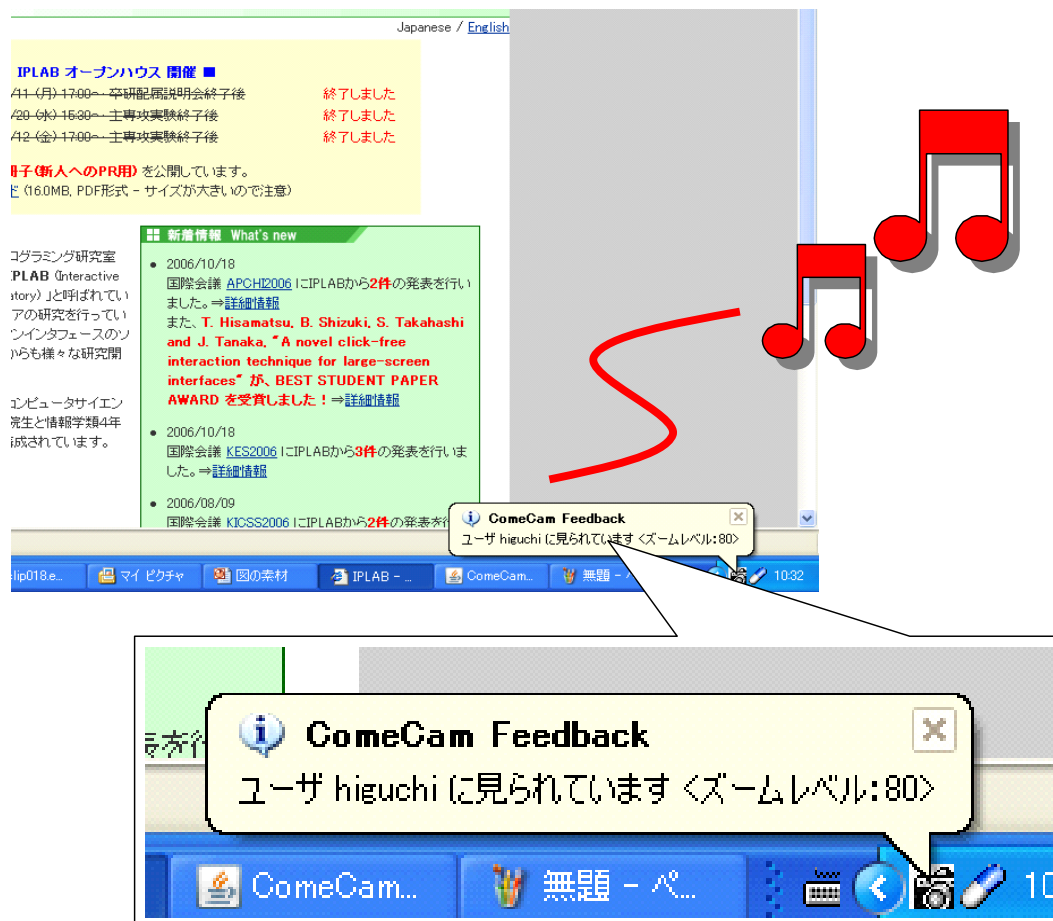


図 4.6: ポップアップメッセージと音声によるフィードバック

4.3.2 フィードバック判定

カメラ視野内のユーザがフィードバック対象になるかどうかの判定を行うのは、「毎 P.T.Z 後」と「ログイン直後」である。前者はカメラの視野が変化する瞬間であり、新たにフィードバック対象となるユーザがカメラ視野内に登場する可能性がある。後者はカメラ画像を始めて見る瞬間であり、すでにフィードバック対象ユーザにフィードバックが行われた後でも、まだカメラ視野内に留まっているのであれば、新たにその対象ユーザをモニタするユーザが出現する可能性がある。この2つのタイミングでフィードバック判定を行い対象ユーザにフィードバックをすることで、見られている状態を確実に気づくことができ、「勝手に見られている」という状況は排除することができると考えられる。

フィードバック判定法の一例

判定の際にはカメラの P.T.Z のため視野が固定されず、ユーザ位置は一意に画像内の (x, y) 座標では特定できないことに留意しなくてはならない。判定法の一つとして、監視対象となる全てのユーザ毎に、カメラの **P.T.Z パラメータ（パン値・チルト値：degree、ズームレベル：整数による相対的な指標値）** に対応したフィードバック必要領域を割り当てるものが考えられる。なお P.T.Z パラメータはカメラにクエリを出すことによって取得できる。

まず (x, y) がある時点でのカメラ画像内の中心座標を示すものとし、これを**カメラの視線の参照座標**と呼ぶ。カメラの視線の参照座標は、ある時点でのカメラのパン値・チルト値、および現時点でのカメラのパン値・チルト値によって一意に特定される。例えばパン値・チルト値がそれぞれ $p1, t1$ のときカメラの視線の参照座標は現在のカメラ画像内で $(x1, y1)$ 、同様に $p2, t2$ のときは $(x2, y2)$ 、といった具合である（つまりパン値・チルト値がそれぞれ $p1, t1$ のときは、現在の画像内と言うところの $(x1, y1)$ の地点が中心に位置するような画像が表示される）。ここでユーザ A のフィードバック必要領域が、現在のパン値・チルト値がそれぞれ P, T であるときの画像内において、 $(x1, y1), (x1, y2), (x2, y1), (x2, y2)$ ($x1 < x2, y1 < y2$) 内の矩形領域であるとする、現在のカメラの視線の参照座標 (X, Y) （すなわち現在のカメラ画像の中心座標）がその領域内にある場合、ユーザ A にフィードバックを行う。以上の説明を模式的に表したものを図 4.7 に示す。

しかしこの方法ではユーザ毎に領域を設定するという手間が発生する。しかもその設定法は、実際にカメラを起動して領域の四隅にパン・チルトし、その時点でのカメラのパン値・チルト値を取得、といったカメラ画像を利用した手作業で、非常に煩雑である。またカメラ位置を移動した際にその設定を一から修正する必要があり、効率的でない。本システムではユーザとカメラの位置を 3 次元空間上で保持し、カメラの視線を検出するアルゴリズムを利用する。

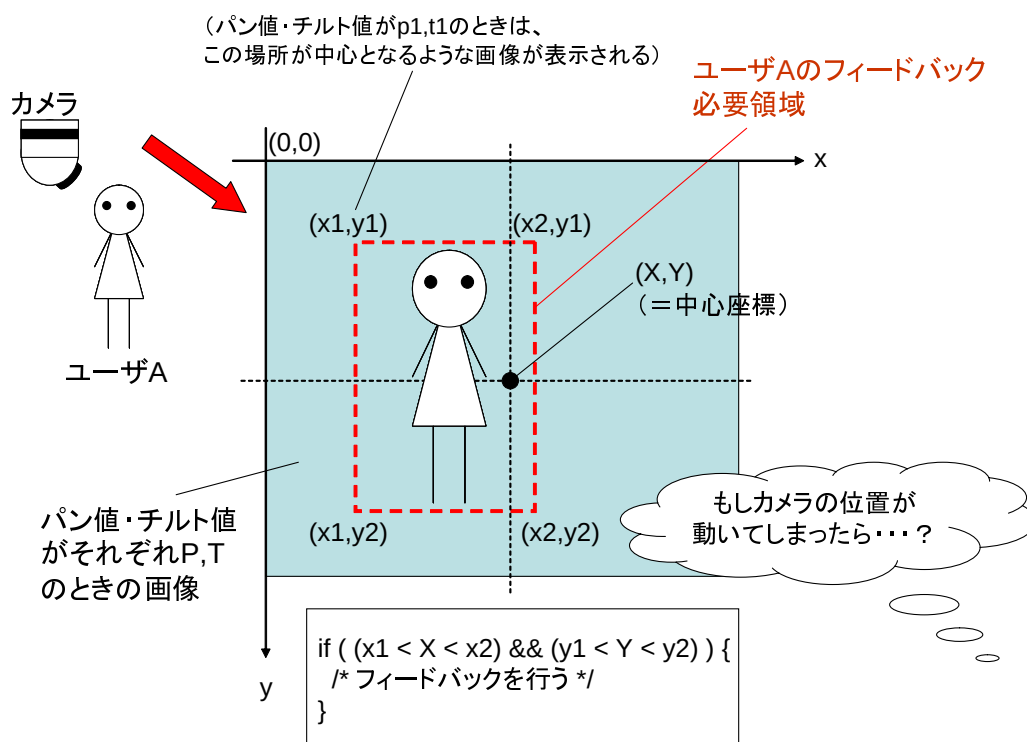


図 4.7: フィードバック判定法の一例

3次元空間でのユーザ・カメラ位置と視線ベクトル

本システムにおいては、図 4.8 のようにユーザやカメラの位置を、部屋の大きさに対する相対的な 3 次元座標で設定する。部屋の大きさやユーザ・カメラ位置は物理的に計測する必要がある。しかし詳しくは後述するが、一度設定してしまえばカメラ位置の移動にも柔軟に対応できる。なお、本システムにおけるユーザの位置は、ユーザのデスクの位置を参考に設定し、動き回らないものとする。

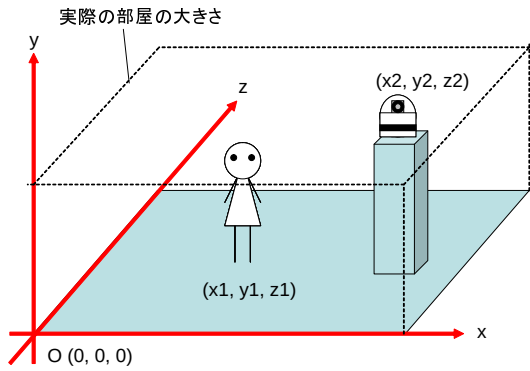


図 4.8: 3 次元空間でのユーザ・カメラ位置

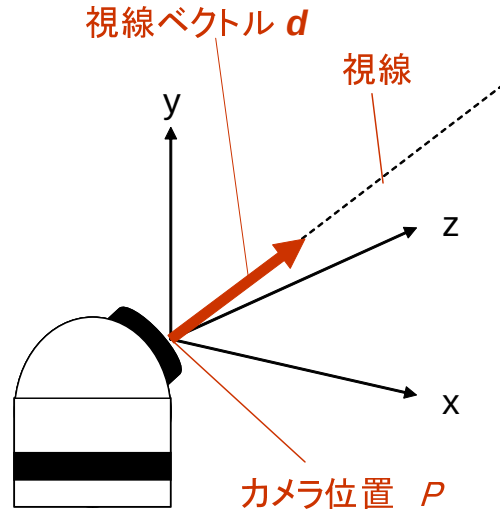


図 4.9: カメラの視線ベクトル

次にカメラの視線を、カメラ位置 P を始点とした無限長の半直線とみなす。またカメラの視線方向を表す長さ 1 のベクトル d を**視線ベクトル**と呼ぶ（図 4.9）。視線ベクトル d は取得した現在のカメラの P.T.Z パラメータのうちパン値・チルト値をもとに、基本ベクトル（参照座標： $(1, 0, 0)$ ）のオイラー回転により算出する。以下に算出式を示す。

$$q = Tp \text{ (} p\text{:元座標, } q\text{:回転後座標) } \theta\text{:パン値, } \phi\text{:チルト値}$$

$$\begin{aligned} p_z &= \begin{bmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} : z \text{ 軸周りの回転} \\ &= (\cos \phi, \sin \phi, 0) \\ &= (x_z, y_z, z_z) \end{aligned}$$

$$\begin{aligned}
p_y &= \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} x_z \\ y_z \\ z_z \end{bmatrix} : y \text{ 軸周りの回転} \\
&= (x_z \cos \theta + z_z \sin \theta, y_z, -x_z \sin \theta + z_z \cos \theta) \\
&= (\cos \phi \cos \theta, \sin \phi, -\cos \phi \sin \theta)
\end{aligned}$$

視線検出アルゴリズム

ユーザのフィードバック必要領域を、ユーザ位置 C を原点とした半径 r の球体として定義する (r の大きさは適当に設定する)。カメラの視線 (半直線) がユーザのフィードバック領域 (球体) と 1 点以上の交点を持つとき、「カメラの視線がユーザの方向を向いている」、すなわち「カメラがユーザを見ている」と判断され、ユーザにフィードバックが行われる。図 4.10 にその模式図を示す。また半直線と球体の交点判別のための計算式を以下に示す。計算式の導出は [13] を参考にした。

$$\begin{aligned}
\|X - C\|^2 &= r^2 : \text{中心座標 } C、\text{半径 } r \text{ の球の方程式} \\
\|P + td - C\|^2 &= r^2 \\
\|(P - C) + td\|^2 &= r^2 \\
(td + (P - C)) \cdot (td + (P - C)) - r^2 &= 0 \\
t^2(d \cdot d) + 2t(d \cdot (P - C)) + (P - C) \cdot (P - C) - r^2 &= 0 \\
t^2 + 2t(d \cdot (P - C)) + (P - C) \cdot (P - C) - r^2 &= 0
\end{aligned}$$

$$t = -d \cdot (P - C) \pm \sqrt{(d \cdot (P - C))^2 - ((P - C) \cdot (P - C) - r^2)}$$

2 次方程式の解の判別式より、 t が解をもつとき線分と球体は交点をもつ

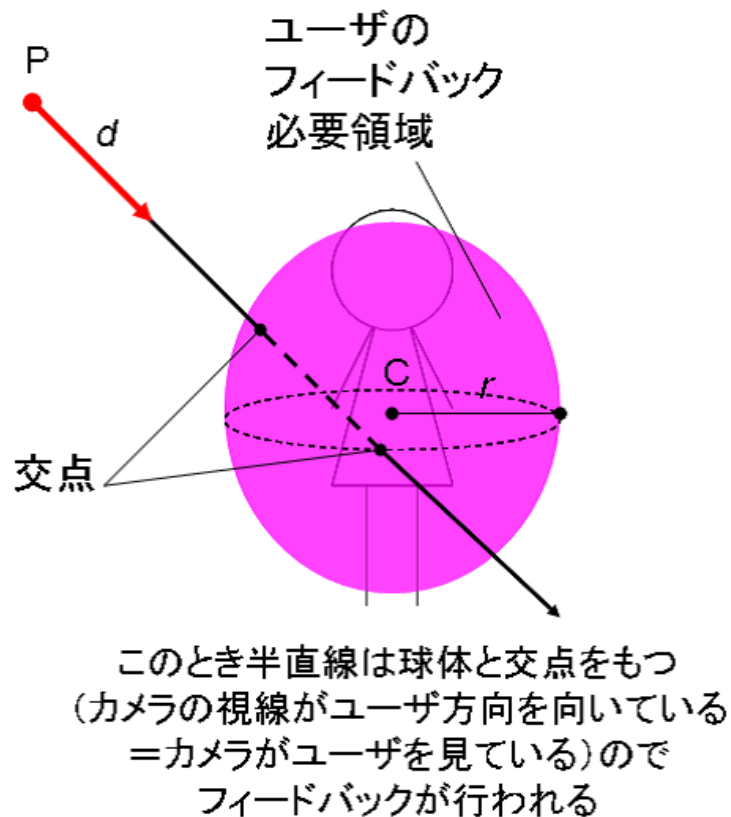


図 4.10: フィードバック判定

以上のようなフィードバック判定法を用いることにより、一度ユーザ位置を設定してしまえばカメラ位置を動かしても柔軟な対応ができる。さらにフィードバック必要領域を球としているため、ユーザ位置と半径さえわかればそれを特定でき、図 4.7 の矩形領域の設定のような手間は必要ない。

これまでフィードバック判定法について説明をしてきたが、本システムではフィードバック判定においてズームレベルを考慮しない。なぜならカメラの視線がユーザ方向を向いているということは、ズームレベルの大きさ如何に関わらず、ユーザを見ていることだと考えられるからである。しかしズームレベルは実際にユーザにフィードバックを行う際に提示される (4.3.1 参照)。つまり本システムでは「ズームレベルが〇〇以下だからフィードバックは行わない」という判定は行わず、ユーザにズームレベルを提示し、その後の判断はユーザに委ねるという手法をとっている。

第5章 マーカーを用いたジェスチャによるカメラの直接的な操作

ComeCam は第3章で提案したような、見られていることを気づくことから誘発されるであろうコミュニケーションを支援し、ネットワークカメラを介した遠隔地における同期型双方向コミュニケーション達成のためのインタラクション手法を、具体的に実現する機能を有する。

第4章で紹介した ComeCam Viewer によるビデオチャットのような対話型コミュニケーションもその一つと言えるが、本章では見られている側からの能動的アクセス手法を実現する機能について解説する。

5.1 マーカーを用いたジェスチャ

本システム：ComeCam はモニタされているユーザがカメラに対して能動的にアクセスし、カメラあるいはカメラ画像を直接的に操作することを可能とする。アクセス手法としてはカメラに対する直接的なジェスチャを用いるが、本システムでは特にマーカーを用いたジェスチャを採用する。つまりここでのジェスチャとは、図 5.1 のようにマーカーをカメラに提示しながら特定のアクションをとることである。

ネットワークカメラ(キャプチャ画像からマーカーを認識する)



図 5.1: マーカーを用いてジェスチャをしている様子

5.2 マーカーの検出

本システムではマーカーの検出に ARToolKit¹ を使用している。ARToolKit とは拡張現実感 (AR:Augmented Reality) の実現を容易にするためのライブラリ群で、ビデオカメラと図 5.1 下のようなビジュアルマーカーを用いることで、図 5.2 のように映像上のマーカーに CG を重ね合わせ、簡単に AR アプリケーションを構築することができる。ARToolKit はビジュアルマーカー領域とマーカー内の画像パターンを読み取り、ビジュアルマーカーやカメラ位置の計算アルゴリズムおよびグラフィックの生成等を行ってくれる。

本システムでは ARToolKit を使用してマーカーを検出し、3 次元空間においてカメラに対するマーカーの相対的な (x, y, z) 座標 (カメラ位置を $(0, 0, 0)$ とする) を取得している。

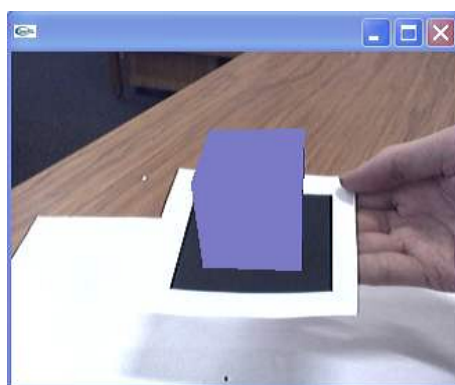


図 5.2: ARToolKit を用いたアプリケーション例

5.3 カメラを呼び寄せる

ユーザはジェスチャを行うことで、指定位置をカメラの視線の参照座標 (カメラ画像の中心座標) となるようなパン・チルト、ならびに一定レベルのズームインを行い、カメラを呼び寄せることができる。

5.3.1 ジェスチャと認識アルゴリズム

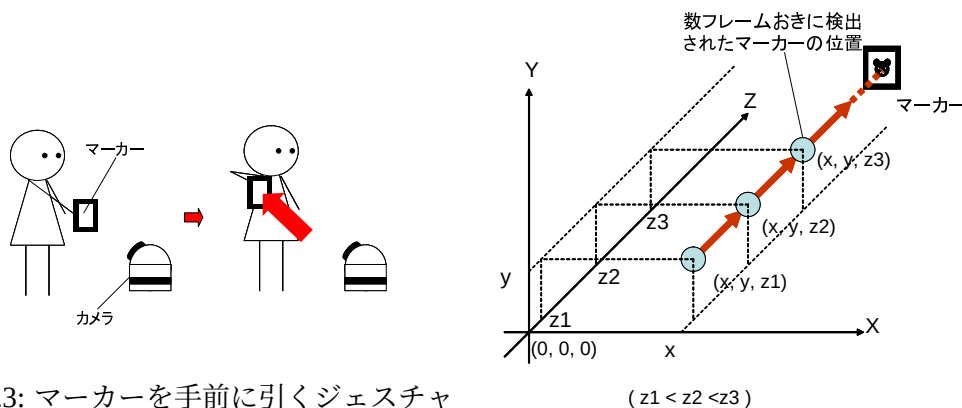
本動作処理においては、図 5.3 のようにマーカーをカメラに対して手前 (ユーザ方向) に引くというジェスチャを利用する。

マーカーの (x, y, z) 座標は一定フレーム毎に検出される。本ジェスチャは「**3次元空間上のマーカーの x, y 座標が閾値以上変化せず、 z 座標が前回検出されたそれよりも大きい**」という条件を一定回数満たすことによって認識され、カメラの P.T.Z 処理が実行される。ジェス

¹ARToolKit Home Page <http://www.hitl.washington.edu/artoolkit/>

チャが認識される際のマーカーの動きの模式図を図 5.4 に示す。つまりある程度の手ブレを許容しながら、マーカーを単純に z 軸方向に引けばよい。

なお、後述するモニタ拒否ジェスチャ (5.4) においては複数のマーカーを使い分けるが、本ジェスチャはマーカーの画像パターンには依存せず、全てのマーカーにおいて同一のジェスチャで認識がされる。



5.3.2 カメラの P.T.Z 処理

パン・チルトは、ジェスチャが認識された時点でのマーカーの位置が、カメラの視線の参照座標となるように実行される。先に述べたように、検出されるマーカーの (x, y, z) 座標は、3次元空間においてカメラ位置を $(0, 0, 0)$ としたときの相対的な座標として示される。図 5.5 のようにジェスチャが認識された際のマーカー位置を (X, Y, Z) とすると、マーカー位置をカメラの視線の参照座標とするために実行すべきパン値・チルト値はそれぞれ、

$$(\text{パン値}) = \arctan \frac{X}{Z}$$

$$(\text{チルト値}) = \arctan \frac{Y}{Z}$$

である。また、ジェスチャが認識された際に実行されるズームのレベル（倍率）は一意に設定されたものである。その倍率は極端に大きなものではなく、適当な回数だけ本ジェスチャを行うことによりユーザは任意の倍率のズームを可能とする。

実際にジェスチャを行ってる様子と、それに伴う ComeCam Viewer 上での画像を図 5.6 に示す。ジェスチャの認識によりズームが実行されている。なお、本ジェスチャと同様にマーカーをカメラ方向に押し出すというジェスチャを行うと、カメラのズームアウトを実行させることができる。

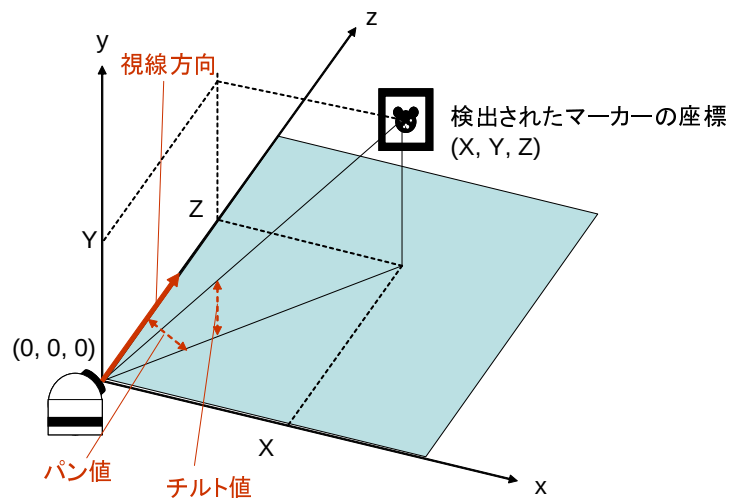
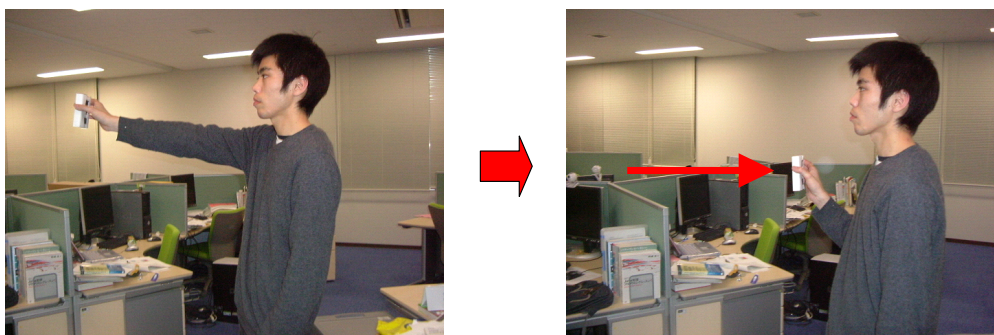
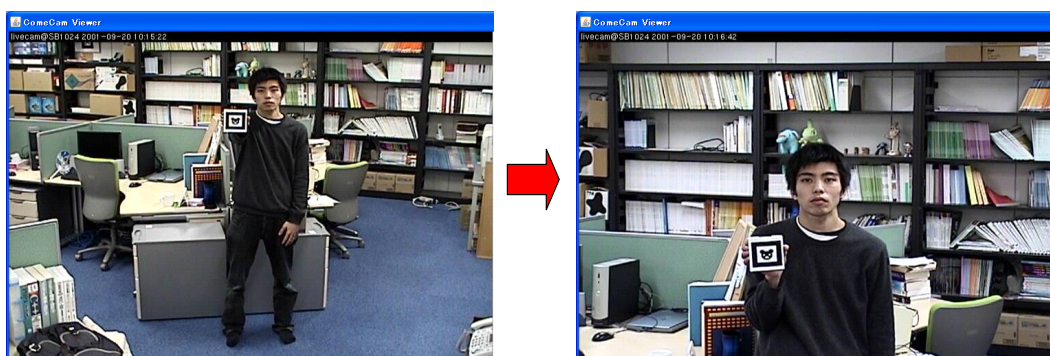


図 5.5: 3次元空間におけるジェスチャが認識されたときのマーカ位置



マーカを手前に引くようなジェスチャを行っている様子



ComeCam Viewer上でのカメラ画像(ズームが実行されている)

図 5.6: カメラを呼び寄せるジェスチャ

5.4 モニタを拒否する

ユーザはジェスチャを行うことで指定位置に任意の大きさのマスクをかけ、モニタを拒否することができる。

5.4.1 ジェスチャと認識アルゴリズム

本動作処理においては、円を描くようにマーカーを動かす（すなわち、マーカーで円を描く）というジェスチャを利用する。なおこのジェスチャは取得されるマーカーの (x, y, z) 座標をもとに、2次元平面上で扱われる。まず「マーカーの x, y 座標が閾値以上変化する」という条件を満たしたとき、現在のマーカー座標 $P1(x1, y1)$ とマーカーの初期座標 $P0(x0, y0)$ を結ぶ線分 $l1$ が作成される。その後同条件を満たすようにマーカーを動かすと、その際のマーカー座標 $P2(x2, y2)$ と $P1(x1, y1)$ を結ぶ線分 $l2$ が作成される。以後はその処理を繰り返す（図 5.7）。なお線分 l は一定数しか保持されず、一定数を超える l が作成されると古いものから順に削除される。

円を動かすようにマーカーを動かすと、線分が交差する。その際の交点と、交点から一番遠い距離にある点 P とを結ぶ直線を直径とする円を作成する（図 5.8）。このとき円の中心座標をマスクの位置、円の大きさ（面積）をマスクの大きさの指標（すなわち作成された円の大きさにマスクの大きさが依存する）と認識し、マスクの描画を行う。なお、線分が交差した際、円を構成する線分 l が一定数に満たないと正しい円が形成されないと考え、そのような場合においては円は作成されない。

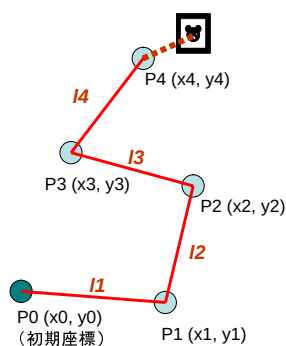


図 5.7: マーカ位置を線分で結ぶ

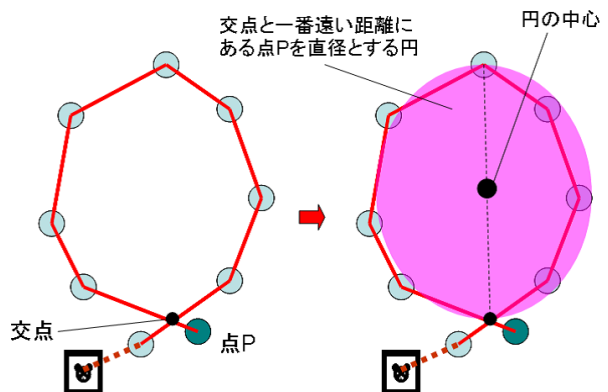


図 5.8: マスクの位置と大きさの特定

図 5.9 は実際にジェスチャを行ってる様子とマスクが生成された様子を、ComeCam Viewer 上での画像を用いて示したものである。デスク上の作業領域にマスクをかけるため、デスク上を囲む円を描くようにマーカーを動かし、球体状のマスクを生成した様子が見えるだろう。



図 5.9: モニタを拒否するジェスチャ

5.4.2 マスクの種類

本システムではマスクとして OpenGL² で作られた 3D オブジェクトを用いており、レンダリングしたオブジェクトをカメラ画像にオーバーレイしている。3D オブジェクトをマスクとして利用する理由は 5.5 で後述する。現在 *ComeCam* では 4 種類のマスクオブジェクトが使用でき、それぞれのマスクオブジェクトはジェスチャに使用するマーカースを使い分けることで切り替えが可能である。図 5.10 に使用可能なマーカースのデザインパターンと、各マーカースに対応して実際に生成された 3D マスクオブジェクトを示す。球体や立方体は形状がシンプルかつ基本的なマスクオブジェクトである。人物以外にもデスクやディスプレイのような特定の領域にマスクをかけたいと思う場合、これらのオブジェクトを適宜大きさを変えて使用するのが望ましいだろう。クマ、およびクマ頭部はユーモア性に特化したマスクオブジェクトであり、マスクでありながら見た目に楽しい画像を提供するので、ユーザのマスクに適している。

5.4.3 マスクの削除および状態変化

現在の実装ではマーカースパターン 1 種類につき 1 つのマスクオブジェクトを作成できる。マスクオブジェクト作成後、同じマーカースでマスクを作成すると同様のジェスチャを行う毎に、マスクの状態を変化させることができる。図 5.10 で示したマスクオブジェクトのうち、球体と立方体はその透明度が段階的に高くなる。これはマスクの公開度に対応するもので、透明度を高くすることによって、完全にマスクするのではなく、なんとなくはマスク領域が見える画像が生成される。クマ、およびクマ頭部は透明度は変化しないが、*y* 軸を中心に段階的に回転する。これはユーザの望む方向にマスクの向きを対応させ、拡張現実感の観点から自然な画像を生成するためである。どのマスクオブジェクトもジェスチャ認識毎に状態変化を繰り返し、ある段階まで変化した後、削除される。

今後は一度マスクが生成された後にマスクの位置を保持し、同一のマーカースで複数のマスクオブジェクトを生成する手法を実装していくつもりである。そうした場合、状態変化および削除を行いたい際には、そのマスク位置を囲むように円を描くジェスチャを行うことによって、対象マスクの状態変化および削除を行う方法が考えられるだろう。

²SGI - OpenGL: Home Page <http://www.sgi.com/products/software/opengl/>

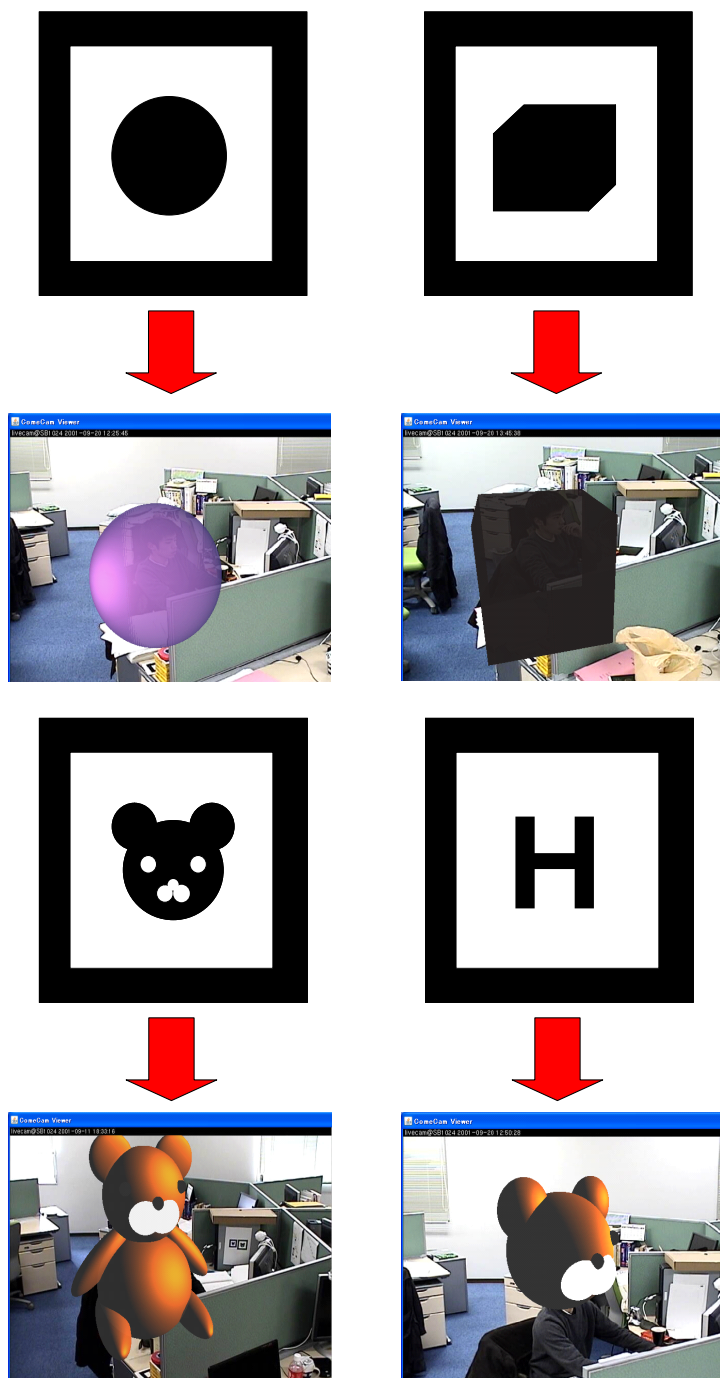


図 5.10: マーカーに対応した 3D マスクオブジェクト（左上から球体、立方体、クマ、クマ頭部）

5.5 カメラの P.T.Z に対応した 3D マスクオブジェクトの生成

モニタ拒否の方法として、指定領域にマスクをかけたり、モザイクなどの画像処理を行う方法は極めて有効と考えられる。しかしカメラが P.T.Z を行う場合、2 次元画像上でのマスク対象領域は一意的に固定されない。対応策としてカメラの P.T.Z 毎にカメラパラメータに対応して図 5.11 のように指定領域を算出し直すという手法も考えられるが、手間の煩雑さから効率的とは言えない。本システムではカメラの P.T.Z に対応したマスクとして 3D オブジェクトを利用する。

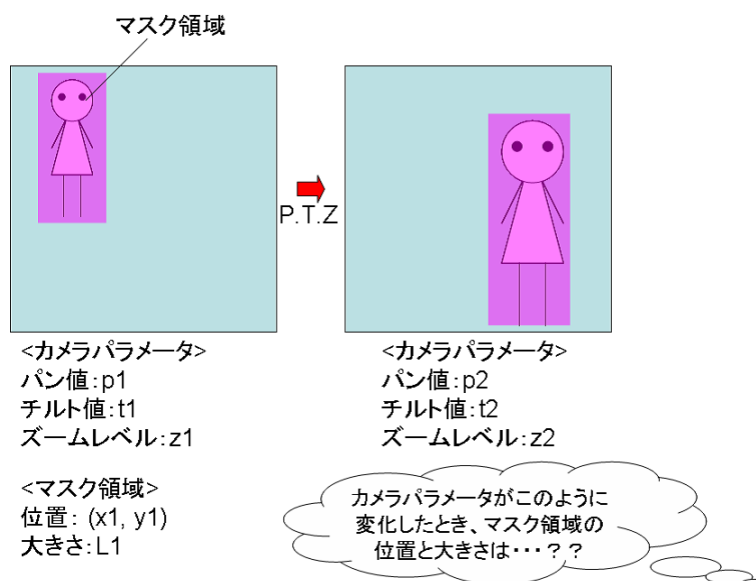


図 5.11: カメラの P.T.Z への対応策の一例

5.5.1 ズームへの対応

本来 3 次元図形をディスプレイなどの 2 次元平面状に表示するためには、3 次元図形を 2 次元図形に変換する処理が必要であり、この処理を投影と呼ぶ。図 5.12 は透視投影法と呼ばれる投影法の一つである。透視投影においては 3 次元空間内にカメラや目に相当する視点を置き、その前方に図形を投影するための投影面を置く。図形が投影する範囲は、前方クリップ面と後方クリップ面で構成されるビューボリュームという四角錐台の領域であり、ビューボリューム外のオブジェクトは投影されない。

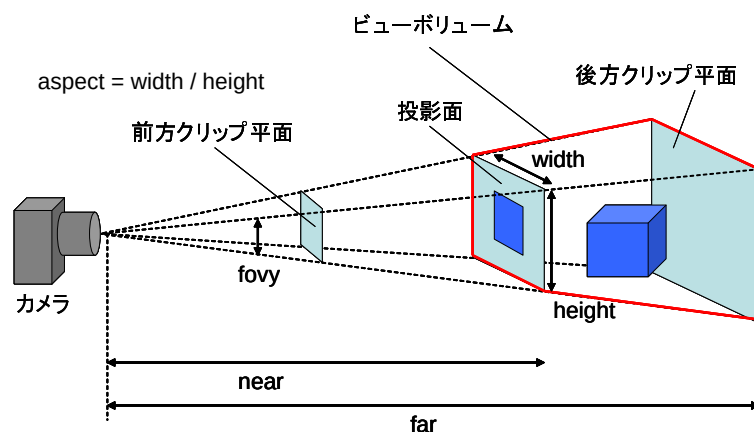


図 5.12: 透視投影法

ディスプレイに表示されるオブジェクトの大きさは、視点から投影面をカバーする角度に依存する。これを画角と呼び、図 5.12 においては fovy に相当する。図 5.13 のように画角が大きいとオブジェクトは投影面に対して小さく写り（広角レンズと同じ効果）、図 5.14 のように画角が小さいとオブジェクトは投影面に対して大きく写る（望遠レンズと同じ効果）。なお図 5.13 と図 5.14 は [14] を参考とした。

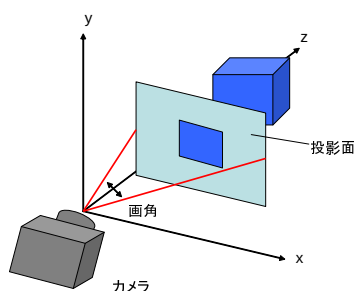


図 5.13: 画角が大きい場合

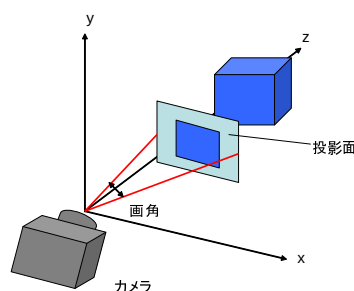


図 5.14: 画角が小さい場合

以上よりカメラのズーム処理が実行された際、3D オブジェクトの透視投影における画角をカメラの光学レンズの画角に対応させることにより、カメラのズームレベルに対応した 3D マスクオブジェクトが常に作成されることになる。

デジタルカメラは CCD とレンズの間の焦点距離を変化させることによって広角・望遠を切り替えるわけだが、例えば AXIS 214 PTZ ネットワークカメラは技術仕様により水平画角が $2.7^{\circ} \sim 48^{\circ}$ と決まっている。これにカメラにクエリを出すことによって得られる現在のカメラのズームレベル (0000～9999) を対応付けることによって、現在のカメラの画角が算出できる。AXIS 214 PTZ の光学ズームの限界値はズームレベルにして 7723 であるから (7724～9999 はデジタルズームの範疇)、画角の算出式は以下のようになる。

(現在のカメラの画角) = $-(48 - 2.7)/7723 \times (\text{現在のカメラのズームレベル}) + 48$

一方 OpenGL には透視投影変換の設定関数として gluPerspective がある。

void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble near, GLdouble far)

引数は図 5.12 中のものと対応している。本システムではカメラのズーム処理毎に現在のカメラの画角を算出し、gluPerspective における fovy 値に代入してオブジェクトを再描画しており、カメラのズームレベルに対応した視野体系の元に 3D マスクオブジェクトが描画される。図 5.15 はカメラをズームした際の ComeCam Viewer 上での画像であり、ズームに対応してマスクも大きくなっているのがわかるだろう。

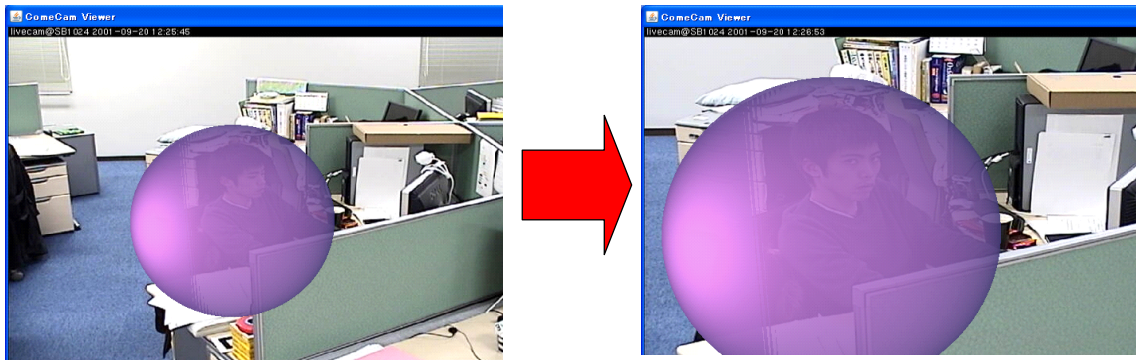


図 5.15: ズームレベルに対応して描画される 3D マスクオブジェクト

5.5.2 パン・チルトへの対応

OpenGL には視点を設定する関数として gluLookAt がある。

**void gluLookAt(GLdouble eyex, GLdouble eyey, GLdouble eyez,
GLdouble centerx, GLdouble centery, GLdouble centerz,
GLdouble upx, GLdouble upy, GLdouble upz)**

(eyex, eyey, eyez) は視点の位置座標、(centerx, centery, centerz) はカメラの視線の参照座標、(upx, upy, upz) は視界の上方向のベクトルをそれぞれ表す。つまりカメラのパン・チルト処理が実行された際、視点位置を 3 次元空間内のカメラ座標に、カメラの視線参照座標を視線ベクトルに対応させることにより (視線ベクトルに関しては 4.3.2 参照)、カメラのパン・チルトに対応した 3D マスクオブジェクトが常に作成されることになる。対応の仕方については図 5.16 を参照して頂きたい。

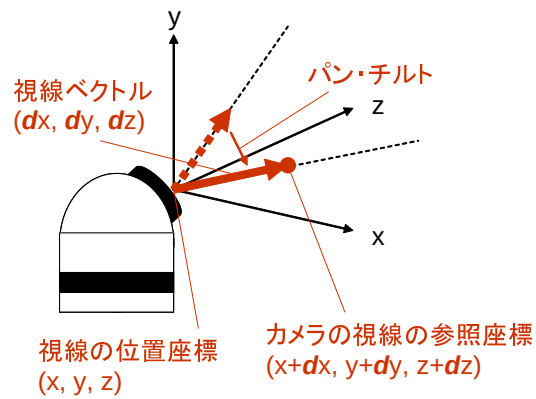


図 5.16: パンチルトへの対応

本システムではカメラのパン・チルト処理毎に現在のカメラの P.T.Z パラメータから視線ベクトルを算出し、gluLookAt における (centerx, centery, centerz) に代入してオブジェクトを再描画しており、カメラのパン・チルトに対応した視野体系の元に 3D マスクオブジェクトが描画される。図 5.17 はカメラをパン・チルトした際の ComeCam Viewer 上での画像であり、パン・チルトを実行してもマスクは同じ位置に保持されているのがわかるだろう。

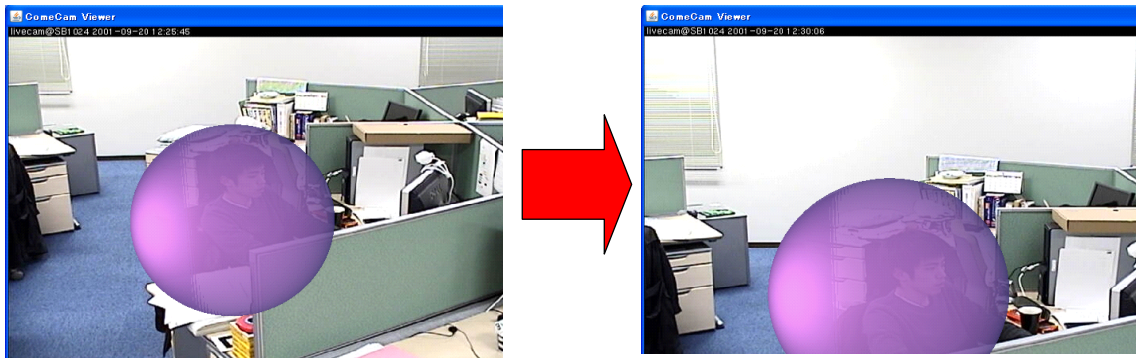


図 5.17: パン・チルトに対応して描画される 3D マスクオブジェクト

第6章 システムの実装

本章では提案・試作したシステムである *ComeCam* の実装、および各動作処理について解説する。

6.1 システム構成

図 6.1 に *ComeCam* システム構成図を示す。本システムは複数台のネットワークカメラ及びユーザ側のクライアント PC、そして *ComeCam MainServer* と呼ばれるサーバー PC にて構成される。各ユーザ側のアプリケーション (*ComeCam Viewer* と *ComeCam Feedback*) と *ComeCam MainServer* 間の各種データのやりとりは、ソケット通信を用いて実行される。

なお、本システムは Java 1.6.0 環境にて実装されている。

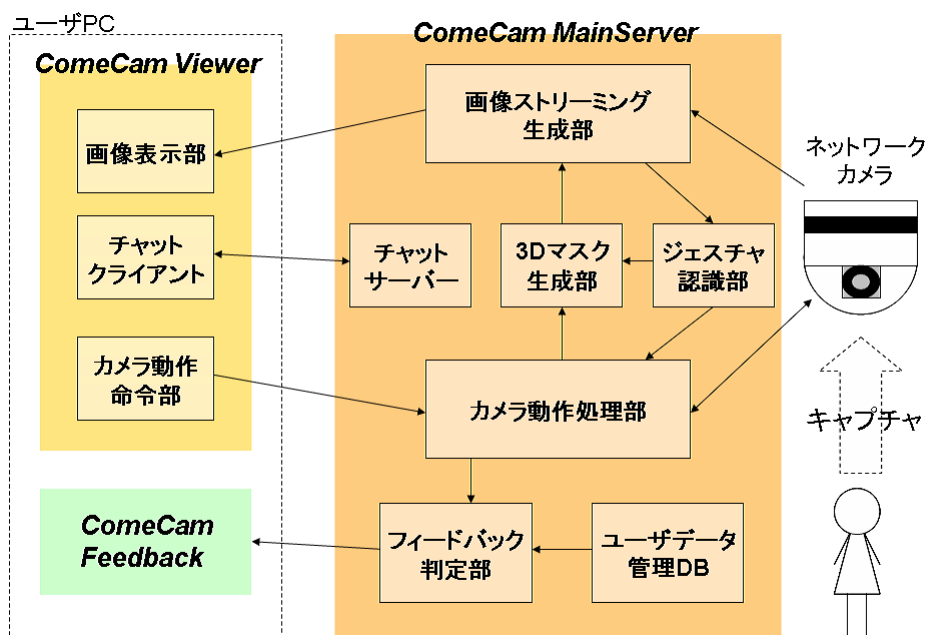


図 6.1: *ComeCam* システム構成図

6.1.1 ネットワークカメラ

本システムではネットワークカメラに図 3.1 の AXIS 214 PTZ¹ を利用した。以下に主な技術仕様を示す。

- ビデオストリーミング：Motion JPEG
- フレームレート：最大 30 フレーム/秒
- 解像度：最大 704 × 480、最小 176 × 120
- レンズ：電動 18 倍光学ズーム、12 倍デジタルズーム (F1.4 f=4.1~73.8mm)
- パン：-170°~170° (最大スピード：100°/秒)
- チルト：-30°~90° (最大スピード：90°/秒)

また AXIS 214 PTZ は HTTP ベースの外部アプリケーションプログラミングインタフェース (API) が利用可能であり、以下のような URL にアクセスすることでカメラに動作命令を出したり、カメラのパラメータを取得したりできる。

- **http://(サーバ名)/mjpg/(カメラ番号)/video.mjpg**
Motion JPEG 形式のデータストリームを取得する。
- **http://(サーバ名)/axis-cgi/com/ptz.cgi?(パラメータ)=(値)[& (パラメータ)=(値)...]**
カメラの P.T.Z を行う。(パラメータ) は動作の種類 (P.T.Z) であり、複数の動作処理を同時 (あるいは順次) 行うことが可能である。
また (パラメータ) = “ center ”、(値) = “ x,y ” でカメラ画像内の (x,y) 座標をカメラの視線の参照座標となるようにパン・チルトすることができる。
さらに (パラメータ) = “ query ”、(値) = “ position ” で現在のカメラパラメータを取得できる。

6.1.2 ComeCam MainServer

ComeCam MainServer は ComeCam システムの主処理やネットワーク管理を行うサーバ PC である。ComeCam MainServer がもつ個々の処理部について説明する。

- 画像ストリーミング生成部：プライバシー保護の観点からマスク処理が施されていない生のビデオストリーミングを、ユーザがカメラから直接取得するという状況は排除する必要がある。ComeCam では本処理部がネットワークカメラからビデオストリーミングを取得し、必要に応じて画像にマスク処理を施した上で、ユーザに配信する。また

¹AXIS: AXIS 214 PTZ <http://www.axiscom.co.jp/prod/214/>

AXIS 214 PTZ から取得された MJPEG (Motion JPEG) ストリーミングは、技術仕様から RTP 通信のようにクライアント PC にそのまま配信することができないため、本処理部は取得した MJPEG ストリーミングから連続的に JPEG 画像を抽出し、独自の画像ストリーミングを生成する。

- チャットサーバー：ユーザが *ComeCam Viewer* 上でチャットを行う際のチャットサーバーとしての処理を行う。
- 3D マスク生成部：3D マスクオブジェクトを生成する。
- ジェスチャ認識部：画像ストリーミング生成部から画像を取得し、マーカの検出ならびにジェスチャ認識を行う。
- カメラ動作処理部：HTTP API を用いてカメラの動作処理を行う。また同様にカメラの P.T.Z パラメータを取得する。
- フィードバック判定部：カメラのパラメータとユーザ管理用 DB におけるユーザ位置を照合して、フィードバック判定を行う (4.3.2 参照)。
- ユーザ管理用 DB: フィードバック判定に参照するユーザ位置や、ログインユーザのデータを格納するデータベースである。具体的には部屋毎に在室するユーザの ID、ユーザ名、ユーザ位置 (3次元座標: (x, y, z))、現在のログインの有無という情報を格納する。

6.2 各動作処理の解説

6.2.1 ログインとチャットシステム

システムにログインするには 4.2 で述べたように *ComeCam Viewer* 起動後、ユーザ名入力フォームに自身のユーザ名を入力する必要がある。入力されたユーザ名は *ComeCam MainServer* に送信される。ユーザ管理用 DB には予めカメラ画像のモニタリングを許可されたユーザデータが格納してあり、*ComeCam MainServer* はそれらと送られてきたユーザ名を照合し、ログインを許可する。ログインが許可されれば *ComeCam Viewer* での画像の閲覧、ならびにチャットシステムの利用が可能となる。また同時に、ユーザ管理用 DB にユーザのログイン情報が格納される。ログインしている全てのユーザには、入力フォームから送信されたテキストが、チャットサーバーを介して送信される。

6.2.2 画像の表示

ログインに成功すると、ComeCam Viewer 上に ComeCam MainServer の画像ストリーミング生成部から取得された画像が表示される。画像ストリーミング生成部から取得するデータは JPEG 画像を符号化したものであり、ComeCam Viewer の画像表示部はそれを連続的に要求ならびに復号化して表示する。各部屋画像の解像度は 704×480 であり、フレームレートはデフォルトで約 10 フレーム/秒である。

6.2.3 カメラの P.T.Z と見られているユーザへのフィードバック

4.2 で述べたように、ComeCam Viewer 上の画像をクリック、あるいは画像上でマウスホイールを動かすことによりカメラの P.T.Z が実行されるが、その動作処理について説明する（図 6.2）。

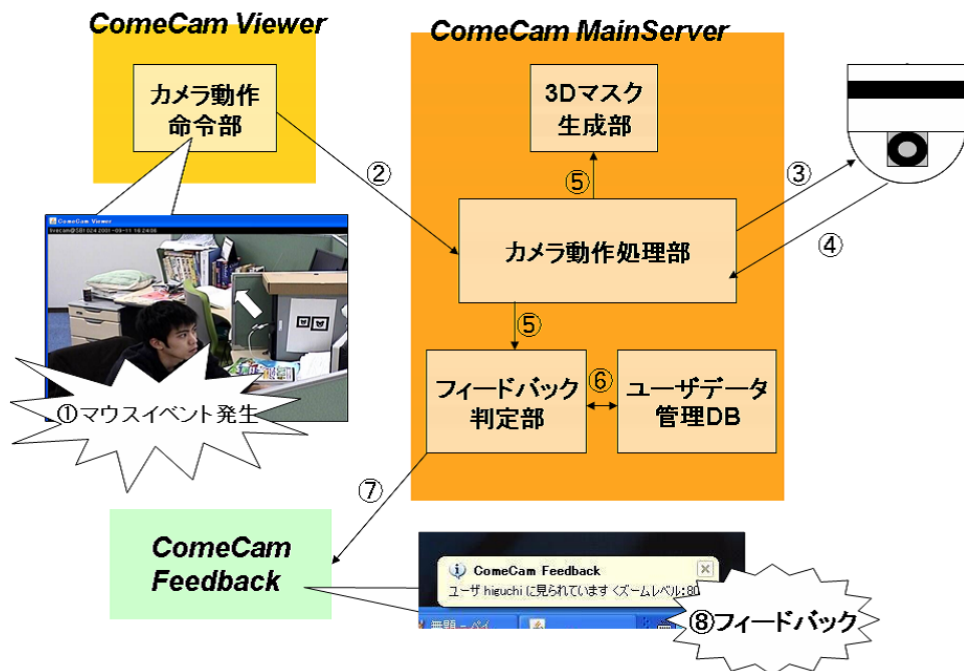


図 6.2: カメラの P.T.Z からフィードバックへの一連の流れ

まず ComeCam Viewer 上でマウスイベントが発生すると (①)、ComeCam Viewer のカメラ動作命令部が ComeCam MainServer のカメラ動作処理部に x, y, z というデータを送信する (②)。このとき x, y はマウスイベントが起こったカメラ画像内の座標、 z はマウスホイールの回転量を示す。カメラ動作処理部は得られた x, y, z を用いて、HTTP API でネットワークカメラの動作処理を行う (③、 z の大きさは実行するズームの倍率に比例)。カメラの動作処

理が終了後、カメラ動作処理部は同じく HTTP API により現在の P.T.Z パラメータを取得し (④)、それらのデータを 3D マスク生成部とフィードバック判定部にそれぞれ渡す (⑤)。

フィードバック判定部は得られたパラメータとユーザデータ管理用 DB を照合し、フィードバック判定を行う (⑥)。フィードバック対象ユーザが特定された際には、対象ユーザの *ComeCam Feedback* にクエリを送信し (⑦)、見られていることへのフィードバックを行う (⑧)。

また 3D マスク生成部は得られた P.T.Z パラメータから、現在のカメラ状態に対応するよう 3D マスクオブジェクトを再描画する。

6.2.4 ジェスチャ認識

ジェスチャ認識における動作処理について説明する (図 6.3)。

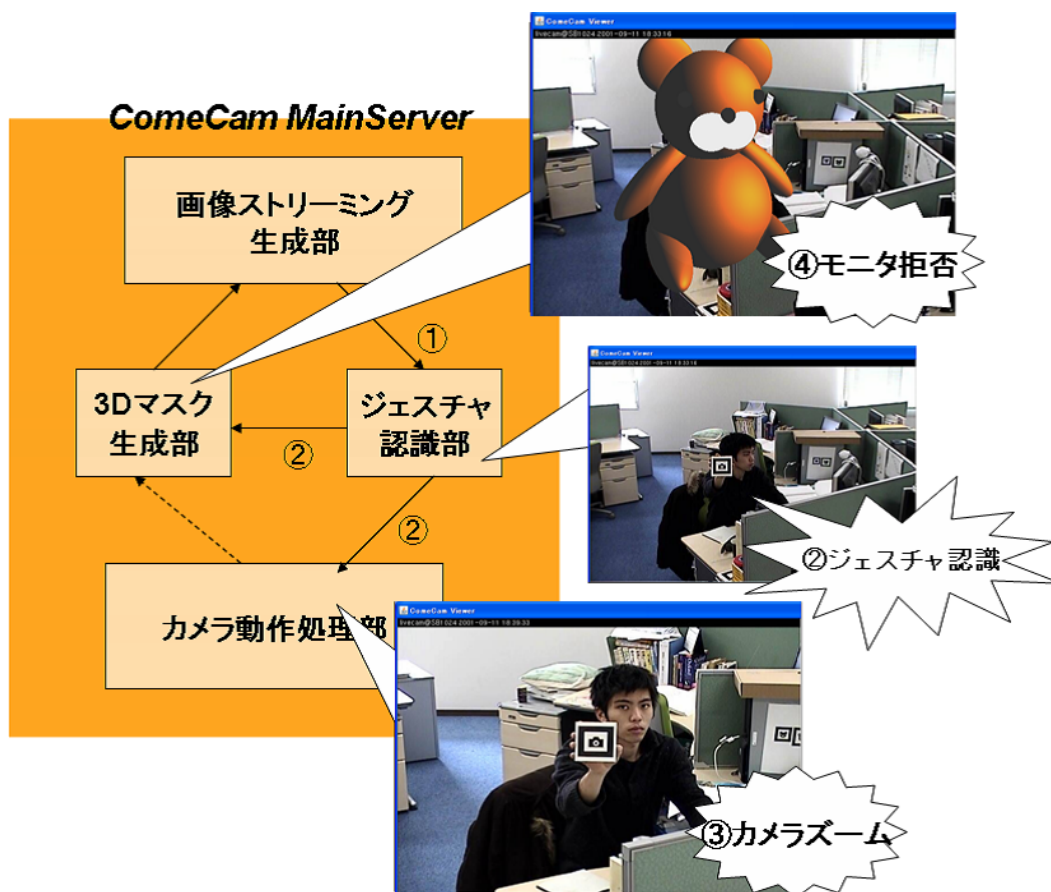


図 6.3: ジェスチャ認識とその後の処理の一連の流れ

ComeCam MainServer のジェスチャ認識部は画像ストリーミング生成部からカメラ画像を取得し (①)、ジェスチャ認識を行う。ジェスチャには ARToolKit によるマーカーを用いるこ

とは前述の通りだが（5.2 参照）、本システムは Java により実装されているため、特に Java 環境での利用を可能にした JARToolKit（Java-Binding ARToolKit）ライブラリを使用する。

画像ストリーミング生成部から取得される画像の解像度は 704×480 であるが、マーカを認識するに当たって解像度がある程度高いとマーカの認識率が損なわれる。しかし新たな解像度の MJPEG ストリーミングをカメラに要求するのは効率的でないため、画像ストリーミング生成部から取得した画像にアフィン変換を施し、解像度を 352×240 まで減じた画像をジェスチャ認識用に利用する。ジェスチャ認識部は変換された画像を整数型のデータ配列に格納し、マーカのデータパターンと照らし合わせる。データパターンと適合した場合、マーカは 3 次元空間上におけるカメラからの相対的な座標 (x, y, z) にて検出される。その後マーカが特定の動きをしたとき、それはジェスチャとして認識され、そのジェスチャに応じて 3D マスク生成部、カメラ動作処理部にそれぞれクエリを出す（②）。

カメラを呼び寄せるジェスチャが認識された際は、 x, y, z データをカメラ動作処理部に渡して適切な P.T.Z.を行わせる（③）。モニタを拒否するジェスチャが認識された際は、マスク生成部にマーカに応じたマスクパターンとマスク位置を送信する。同時にモニタ拒否ジェスチャ認識毎に、マスクの存在を判定する。既存マスクが存在する場合には、既存マスクの状態に応じて状態変化（透明度、方向の変換）、あるいは削除のクエリを 3D マスク生成部に送信し、オブジェクトを再描画させる（④）。

第7章 考察と課題

今回試作した *ComeCam* は、筑波大学の分散された研究室での運用を想定して実装された。この研究室においては以前から日常的にネットワークカメラが稼動しており、それぞれの研究室に在室するメンバーのプレゼンス情報の取得等に頻繁に利用される。カメラの設置は皆が既に了承済みだが、特定の場面においてカメラを意識したり、カメラに写らないような工夫をする光景もよく見受けられた。*ComeCam Viewer* や *ComeCam Feedback* はカメラに対するほぼ確信的な安心感を与えると共に、モニタする側・される側が既知の間柄であることもあり、発展的なコミュニケーションが期待できる。またマーカによるカメラ操作や P.T.Z カメラに対応した 3D マスクオブジェクトの作成は、新鮮でありその行為自体が楽しいものであると感じられた。

しかしシステム運用について考察していく上で、*ComeCam* には今後考慮しなくてはならない幾つかの問題点があることがわかった。

まずフィードバック判定法に関してだが、試作した *ComeCam* は、フィードバック判定におけるユーザ位置を、デスク位置といった固定的なものに限定している。今回は研究室内で運用し、ユーザの活動領域は自身のデスクを中心としたものになるので、問題はなかった。しかし現実的にはユーザ位置は固定的なものでなく、ユーザが動き回るような場合も考えられ、そうしたケースにも対応する必要があるだろう。そうした場合、3次元空間上でのユーザの動きを追跡する技術が必要となるかもしれない。ユーザの動きの追跡に画像のキャプチャと同一のカメラを用いるとするならば、カメラの動作検出機能の応用が可能であると考えられるが、各ユーザの特定には依然として問題が残る。

またフィードバック手法自体にも課題が残る。例えば前述のようにユーザ位置が固定的でない場合、またユーザが PC の付近にいない場合、ディスプレイ上のメッセージや PC からの音声といったフィードバック手法では不十分である（今回のように研究室やオフィス環境での運用だと、ユーザ位置はデスク位置と特定され、さらにそのデスクには PC があることが多いので、非常に都合がよい）。対応策の一つとして、携帯電話のようなモバイルデバイスに対して信号を送るような手法が考えられる。

今回はディスプレイ上でのメッセージと PC からの音声という 2 つのフィードバック手法を実装した。しかしながら当然他のフィードバック手法も考えられる。例えばメッセージや数値（ズームレベル）などの情報で、直接的に見られていることを明示するのではなく、特に注意を払わずともそれらの情報を常時なんとなく感じていられるようにする情報提示装置、いわゆるアンビエントディスプレイのようなものを利用することが考えられる。アンビエント

ディスプレイの例として、ネットワーク上の情報の流量を部屋の天井付近に取り付けた風車の回転速度で提示するというような実装例がある [15]。本研究にアンビエントディスプレイを応用するならば、見られることによって対象ユーザに付随するランプが光り、ズームレベルによって明るさが変化する、といったものが考えられる。そもそも見る側・見られる側の人間が、ネットワーク越しではなく同じ部屋にいる場合に、「見る」「見られたことに気づく」という状態ははっきりと明示されるものではない。本研究は被撮影者のネットワークカメラに「見られること」への抵抗感を緩和することを目的としているが、このような同室感を強く感じさせるような枠組みを考えることは、カメラに対する特別な意識を軽減させることにつながるのではないか。

次にジェスチャによるカメラの直接的な操作に関してだが、マーカー認識を用いるためそもそもカメラ視野内にいないとカメラを呼び寄せたりできないという問題がある。この問題の対応策としては画像をキャプチャするカメラとは別に、遠景でマーカーを検出するためだけに稼動するカメラを設置するという方法が考えられるが、同じ空間に複数のカメラを設置することは本研究の趣旨からしてあまり望ましくない。近年では広範囲を撮影できる広角レンズを有した P.T.Z 対応ネットワークカメラが開発されており、そうしたカメラを利用することは非常に有効であると考えられる。またジェスチャ自体に関しても、[11] のようにマーカーといった一切の道具を用いない手振りのようなジェスチャを認識することができれば、カメラの操作はユーザにとってより直感的になると言える。

またモニタ拒否ジェスチャによって生成されるマスクにも課題が残る。マスクオブジェクトはカメラ画像上にオーバーレイされただけなので、例えばユーザを隠すような 3D マスクオブジェクトを生成したとして、その際ユーザの手前（ユーザとカメラの間）にユーザの一部を隠すしきりのようなものがあつたとする。マスクオブジェクトは単にオーバーレイされるだけなので、常に最前面にあるかのように描画される。ゆえにマスクオブジェクトはユーザの手前ではなく、しきりの手前にあるかのように描画されてしまい、拡張現実感の観点からすると不自然な画像が生成されてしまうのである。そのため見た目に自然な画像を得るためには、先の例で言うとしきりによってマスクの一部が隠れる（すなわち、しきりよりマスクの方が奥にある）かのように描画を行う手法を考案する必要がある。

我々は今後、こうした問題に対応するための具体的な方法を探索・実装するとともに、評価基準を明確に整えた上で、システムに関する客観的な評価を得るための実験を行う予定である。特に先に述べたような適切なフィードバック手法を考案するためには、幾つかの情報提示装置を試作し、実験を通して得られた知見をもとに最適なものを選択する必要があるだろう。

第8章 結論

本研究ではまず被撮影者のネットワークカメラに対する「見られていること」への抵抗感を緩和するための手法として、「見られていることを気づかせる」手法を提案した。「見られていること」に気づくことによって、被撮影者は容易にモニタ拒否などの対応策をとることができる。しかもその対応の判断は被撮影者に委ねられるため、モニタ領域の制限は最小限に抑えることができ、モニタリングカメラとしての役割を損なう可能性は軽減される。同時に我々は「見られていることに気づく」ことから誘発されるモニタする側・される側のコミュニケーションの可能性を考慮し、それを支援するための両者のインタラクション手法を提案した。一つはビデオチャットのような対話型コミュニケーション、もう一つは被撮影者がカメラに対して能動的にアクセスし、カメラ（画像）を直接的に操作する手法である。

次に我々は提案手法を具体的に実現するネットワークカメラシステム：*ComeCam* を実装した。*ComeCam Viewer* は複数の部屋の同時モニタリングやカメラの視野操作、ビデオチャットのような対話型コミュニケーション環境を提供する。*ComeCam Feedback* は被撮影者にモニタしているユーザ名ならびにズームレベルを提示するフィードバックを行い、被撮影者に「見られていること」を気づかせる。また *ComeCam* はカメラに対しマーカーを用いたジェスチャを行うことにより、被撮影者からカメラを呼び寄せたり、画像にモニタを拒否するためのマスクをかけることを実現する。

我々は本研究における提案手法、ならびに *ComeCam* の試作・運用を通して、ネットワークカメラに対する「見られていること」への抵抗感を緩和させ、ネットワークカメラを我々の生活にとって受け入れやすいものにすることができたと考える。同時にネットワークカメラの利用用途に関して、コミュニケーションツールとしての拡張を考案した。これらの成果は今後も増大を続けるネットワークカメラの利用シーンや、ユビキタスコンピューティングの発展に寄与できるだろう。

謝辞

本論文を執筆するにあたり、指導教員である田中二郎先生をはじめ、高橋伸先生、三末和男先生、ならびに志築文太郎先生には、幾度となく丁寧なご指導と適切な助言を頂きました。心より感謝申し上げます。

また田中研究室の皆様にもゼミなどを通じて大変貴重なご意見を頂きました。特にユビキタスチームの皆様には、チームミーティング以外にも日常的に多くのご意見やご指摘を頂きました。本当にありがとうございました。

最後に、家族や友人、サークルの先輩・後輩にも様々な面で支えて頂きました。この場を借りて御礼申し上げます。

参考文献

- [1] Axis. ネットワークカメラとは. Axis White Paper, 2002.
- [2] 大田友一. 計算メディア：センシング機能がネットワークと融合するとき. フェロー&マスターズ未来技術研究会資料, pp9-14, 2006.
- [3] Bent Hamer 監督. キッチン・ストーリー (原題 Salmer fra kjøkkenet). BULBUL FILM AS, 2003.
- [4] Scott E. Hudson, Ian Smith. Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support System. the 1996 ACM conference on Computer supported cooperative work, pp248-257, 1996.
- [5] Michael Boyle, Christopher Edwards, Saul Greenberg. The Effects of Filtered Video on Awareness and Privacy. the 2000 ACM conference on Computer supported cooperative work, pp1-10, 2000.
- [6] Carman Neustaedter, Saul Greenberg. The Design of a Context-Aware Home Media Space. UBICOMP 2003 Fifth International Conference on Ubiquitous Computing, pp297-314, 2003.
- [7] 高橋伸, 岩淵志学, ジャッキーノ ヤン, 山田徹, 久松孝臣, 中村卓, 土持幸久, 金春明, 田中二郎. ライブカメラ画像を用いたプレゼンス情報の表示手法. 第13回インタラクティブシステムとソフトウェアに関するワークショップ WISS2005, pp15-18, 2005.
- [8] Takehiko Ohno. Weak Gaze Awareness in Video-Mediated Communication. CHI 2005, pp1709-1712, 2005.
- [9] 早川祐, 葛岡英明. 遠隔会議における指さし支援の研究. ヒューマンインタフェースシンポジウム 2004 論文集, pp459-462, 2004.
- [10] 橋本悟, 中西泰人. SpaceTracer: ネットワークカメラ画像の合成による空間型コミュニケーションシステムの提案. インタラクシオン 2006 論文集, pp.181-182, 2006.
- [11] 入江耕太, 梅田和昇. 濃淡値の時系列変化を利用した画像からの手振りの検出—全方位視覚センサと赤外線カメラへの適用. 電気学会 電子・情報・システム部門大会, 2002.
- [12] 竹原伸彦, 渡部智樹, 安西浩樹. 遠隔映像視聴のための直感的操作インタフェースのシステム評価. 第2回 情報科学技術フォーラム FIT2003, pp579-580, 2003.

- [13] Philip J. Schneider, David H.Eberly. Geometric Tools for Computer Graphics. Morgan Kaufmann Publishers, 2003.
- [14] 藤代一成他. コンピュータグラフィックス. CG-ARTS 協会, 2004.
- [15] Hiroshi Ishii, Sandia Ren, Phil Frei. Pinwheels: Visualizing Information Flow in an Architectural Space. CHI 2001,pp111-112, 2001.