

平成 15 年度茨城大学情報工学科卒業論文

ピアツーピア型バーチャル・ボールゲーム

提出年月日： 平成 16 年 2 月 10 日

著 者： 茨城大学工学部情報工学科 00T4015R 河野 義広

指導教官： 茨城大学工学部情報工学科助教授 米倉 達広

目次

第1章	はじめに.....	1
第2章	問題解決への基本概念.....	3
2.1	AtoZの導入.....	3
2.2	相補予測.....	6
第3章	AtoZと相補予測の計算方法.....	8
3.1	AtoZの計算方法.....	8
3.2	相補予測の計算方法.....	11
第4章	クリティカル・ケース.....	13
4.1	クリティカル・ケース.....	13
4.2	Dead Zone.....	14
第5章	実験.....	16
5.1	実験タスク.....	16
5.2	実験の目的.....	17
5.3	実験方法と結果.....	18
5.4	考察.....	27
第6章	まとめ.....	29
	謝辞.....	30
	参考文献.....	31
	付録.....	33
1.	ボロノイ領域.....	33
2.	マハラノビスの距離.....	34
3.	2焦点から求める楕円方程式.....	37

第1章 はじめに

近年、インターネットの急速な普及による利用者の急増と通信データの大容量化に伴い、対話型のアプリケーションが普及してきた。それに伴い例えばリアルタイム遠隔操作やネットワーク対戦型ゲームなどの実時間対話性の高いアプリケーションでは通信遅延の問題が顕著となっている[1]-[3]。

通常、分散仮想環境(以下 DVE と略記)では、サーバ・クライアント型トポロジを用いることが一般的である[1]。この型の DVE では、仮想状態の一貫性(consistency)の面で有利ではあるが、サーバへの負荷集中や、サーバを経由したデータ配信によるフィードバック遅れなどが問題として知られている[10]。そこで最近ではネットワーク通信の高速化や、通信データの高圧縮化など個別の対策がとられているが、これらは通信遅延自体の本質的な軽減に寄与するものではない。一方で peer-to-peer(以下 P2P と略記)型の DVE が注目されている。P2P 型 DVE では、個々の peer が自分以外の peer と直接通信を行うため、フィードバックの遅れが少なく済む。しかしながら、各 peer の状態を同期させるのが難しいため、inconsistency の問題が発生する。これは consistency-throughput のトレードオフ問題[10]として知られており、consistency (throughput)を優先するには throughput (consistency)が犠牲となる。つまり、P2P 型 DVE ではローカルな情報は即座に提示できる一方、リモート情報は提示内容(時刻)に差が生じるというように、情報提示時刻の差異(以下、これを情報提示の差異とする)が大きく、共有オブジェクトの扱いが困難であることが知られている[1][3]。また、P2P を用いた大規模ネットワークにおいて更新情報を共有する通信プロトコルの研究等も報告されている[8][9]。しかし、これらは実際のネットワークではなく、計算機シミュレーションのみによる評価であると同時に各アバタ(DVE上の仮想プレイヤー)情報の通信のみが想定されており、実際の共有オブジェクトの扱い方に言及してはいなかった。

そこで本論文では、P2P 型ネットワークでの consistency および一貫性のある共有オブジェクトの管理手法の提案[4][5]を目的とする。本論文では、DVE として仮想の物理空間を想定し、アバタや共有オブジェクトの属性が位置や速度などの物理情報を示すような DVE (フィールド型 DVE とよぶ)を対象とする。P2P のフィールド型 DVE を構成するには共有オブジェクトの管理権(どの peer がオブジェクトを制御する権利を有するか)、ならびに各 peer での物理属性(共有オブジェクトと全アバタの属性)の一貫性が最も重要な課題となる。

そこで

- 1) 共有オブジェクトの管理権がいずれの peer に存在するか (管理者とよぶ) の認識を全 peer が共通に持つことを保証する (管理権の整合性の保証)
- 2) 上記の管理権の決定はできる限り合理的になされなければならない
(管理権の合理的決定)
- 3) 全 peer で全てのアバタの物理属性が一致する (アバタ状態の整合性の保証)
- 4) ネットワーク遅延により上記 1) や 3) の共通認識が時間的にずれてしまうことの軽減 (遅延による影響の軽減)

の 4 項目を考える必要がある。以下、2 章でこのための基本的概念として AtoZ (Allocated Topographical Zone) と相補予測の 2 つを導入し、3 章でそれらの計算法と諸性質を与える。また、4 章で通信遅延によって発生する peer 間のクリティカル・ケースについて対処する。そして 5 章でそれらを利用したフィールド型 DVE として実装した、ネットワーク対戦型エアホッケーにより本論文の提案手法の有効性と適用範囲について実験的な評価を行う。

第2章 問題解決への基本概念

本章では、1章で示した問題解決のための2つの基本的概念について説明する。

2.1 AtoZの導入

ここで与えられた条件は、各 peer にはプレイヤーがいて、それぞれが自分の（ローカルな）アバタのみを直接的に制御できる。共有オブジェクトは絶えず唯一つのアバタにより排他的に管理可能な状態に置かれる。この管理権を有するアバタは絶えず動的に変わり得る。

このような条件下で前節における項目 1) と 2) を実現するために、共有オブジェクトの扱いに関する各 peer 間の分散処理プロトコルを考える。そこでまず、各アバタが優先的に共有オブジェクトを管理できる、空間領域という概念を導入する。これはフィールド型 DVE においてどのプレイヤーが最短時間で共有オブジェクトにアクセスできるかをフィールド全体で一義的に決定するための概念であり、各アバタの位置、速度、進行方向により、他アバタとの間の相互関係を考慮に入れて決定される。本論文ではこれを Allocated Topographical Zone（地形的な割り当て領域）の意味で、以下 AtoZ と呼ぶ（図1）。これは具体的に

$$\begin{aligned} AtoZ(A) &= \{x \mid x \in Z, \min(\forall Access(O)) \\ &= Time(O, A, x)\} \end{aligned} \quad (1)$$

で定義できる。ただし $AtoZ(A)$ はアバタ A に属する AtoZ 領域、 x は位置情報、 Z は領域全体を示している。また、 $\forall Access(O)$ は全てのアバタが共有オブジェクト O にアクセスするまでの時間、 $Time(O, A, x)$ は地点 x にいるアバタ A が共有オブジェクト O にアクセスするまでの最短時間を示している。

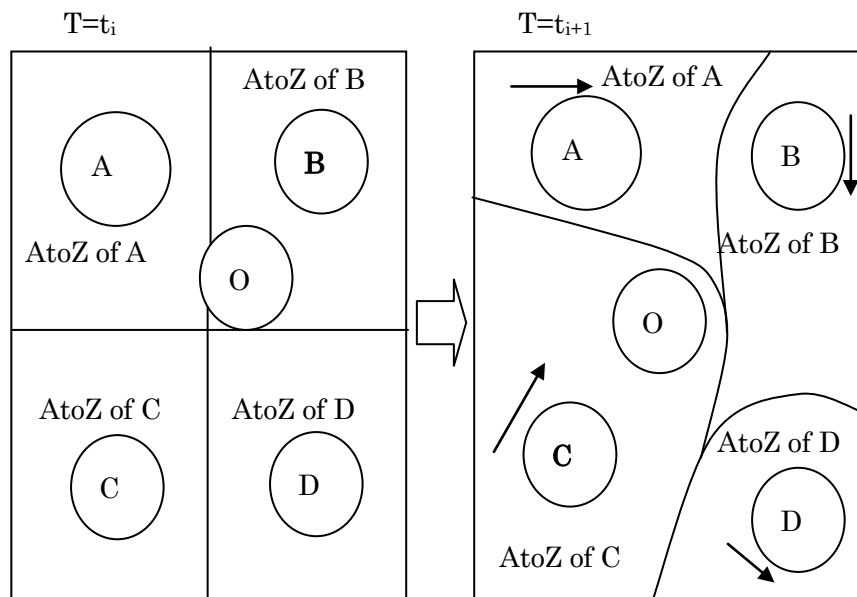


図 1. AtoZ の概要
 Fig.1 The Over-view of AtoZ

図 1 において、A~D はアバタ A~D を示し、O は共有オブジェクトを示す。すると、時刻 $T=t_i$ の時点で、共有オブジェクト O およびアバタ A~D は静止している。このとき O は端末 B の AtoZ 内にあるので、管理者は B となる。次に時刻が $T=t_{i+1}$ ($t_i < t_{i+1}$) に変化した時点では、各 AtoZ がアバタの移動により変化している。このとき O は端末 C の AtoZ 内にあるので、管理者は C となる。

ここで各 peer がそれぞれのアバタの AtoZ を共通に認識するという仮定で、現時点で共有オブジェクト O が位置する AtoZ の属するアバタがその共有オブジェクトの管理権 (共有オブジェクトの情報を制御する権限) を持つようにすれば、管理権の整合性の保証 (項目 1))、ならびにその合理性 (項目 2)) を満たすことができる。ただし各 peer がそれぞれの AtoZ 領域を共通に認識するという仮定が成り立つためには、項目 3) が満たされ、かつ AtoZ の決定方式を全 peer が共有する (計算式とその計算精度が一致する) 必要がある。

項目 3) は 各アバタを制御する各 peer が自分のアバタの物理属性情報を他の全 peer に絶えず送信し続け、自分以外のアバタの物理属性情報を絶えず受信し続ける ことで可能となる。

従ってこれらをまとめると

<P2P での AtoZ による分散処理プロトコル>

①各 peer は自己アバタの状態を他の全 peer に定期的に送信し、他の全 peer から自己以外の各アバタ状態を定期的に受信する。

②各 peer で上記①に基づく仮想空間内の全アバタに属する AtoZ 領域を独自に計算する。

③上記②の結果、共有オブジェクト O が自己アバタに属する AtoZ に存在するなら 3-1) を実行、そうでないなら 3-2) を実行する

3-1) 自分の peer が O の管理権をもつものとして、自己アバタと O の相互作用を計算し仮想空間の状態を更新してその結果を他の全 peer に送信する。

3-2) O が属するアバタの peer から O に関する更新情報を受信し仮想空間の状態を更新する。

の手順をそれぞれの peer が同時並列的に実行すれば良い。ここで具体的な AtoZ の計算法については 3 章で詳細に説明する。また、手順①および③で発生すると考えられる遅延による時間ずれの問題は前章の項目 4) と等価である。これについては以下で述べる相補予測により対処する。

2. 2 相補予測

ここでは、P2P 型の最も簡単な例として 2 つの peer の場合について考察するが、peer 数が 3 個以上でも原理的に同じである。このモデルでは、片方の peer である peer A ともう一方の peer B の 2 者の間でのみデータの送受信が行われる。このとき、前節の protocol により、peer A(peer B)にはローカルに操作するアバタ A(アバタ B)の状態、リモート peer B(peer A)によって操作されるアバタ B(アバタ A)の状態、および共有オブジェクト O の状態が提示されることになる。このような DVE では、peer 間で通信遅延による情報提示の差異が生じてしまう (図 2)。このとき peer A から peer B への通信は Δt_a の遅延が発生し、その逆方向への通信には Δt_b の遅延が発生している。この時点で、共有オブジェクトは AtoZ の計算結果から peer A によって管理されているものとする。

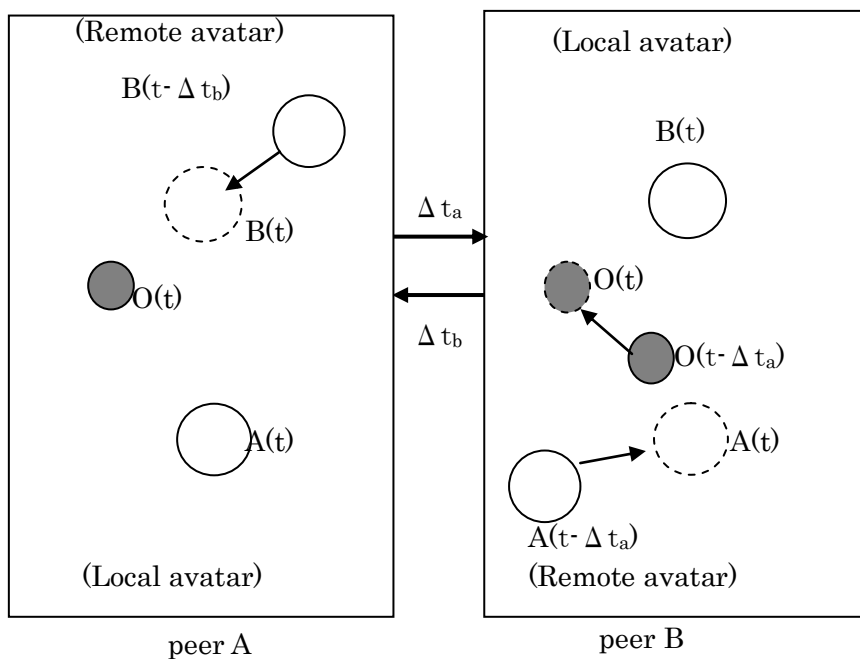


図 2. Peer 間での情報提示の差異

Fig.2 The gap of the displayed information between the two peers

図2において、peer A(peer B)に提示されるアバタ B(A)の最新状態は、遅延分 $\Delta t_b(\Delta t_a)$ を現時刻から差し引いた $t - \Delta t_b(t - \Delta t_a)$ の時点での状態である。また共有オブジェクトの状態は peer B においてのみ Δt_a の遅延を伴って観測される。従って、両 peer に提示される両アバタおよび O の状態時刻は peer A および peer B で以下のように表される。

(peer A での提示情報)

$$A_A(t) = A(t) \quad B_A(t) = B(t - \Delta t_b) \quad O_A(t) = O(t) \quad (2)$$

(peer B での提示情報)

$$A_B(t) = A(t - \Delta t_a) \quad B_B(t) = B(t) \quad O_B(t) = O(t - \Delta t_a) \quad (3)$$

ただし各式の左辺は各 peer に提示されるアバタおよび共有オブジェクトの状態を示し、その添え字は提示される peer を表している。また右辺はその各時刻における実際の（それが制御されている条件下での）状態を示し、括弧内は時刻を表す。このように peer 間で情報提示の差異が生じる場合、これが対話性の劣化を引き起こすことが確認されている[1]。

この問題に対して、本論文では各 peer 上で他のアバタの状態を予測するとともに、自己アバタの状態が他の peer 上でどう予測されているかを相互に予測することで対処する[7]。本論文では、相補予測の結果を AtoZ の計算に反映させている。次章で AtoZ と相補予測の具体的な計算法を与える。

第3章 AtoZ と相補予測の計算方法

本章では、AtoZ と相補予測の具体的な計算方法について説明する。

3. 1 AtoZ の計算方法

ここでは、AtoZ を実際に計算するための方法について説明する。複数のフィールドプレイヤー間で共通のオブジェクト（例えばボール）を優先的に処理できるためのフィールド分割法については例えば、既に瀧・長谷川らにより「優勢領域(dominant region)」とよばれる概念が提案されている[10][11]。これは、チームスポーツの観点からフィールドスポーツにおける「スペース」を定量化したものである。優勢領域とは、一種の動的な勢力範囲であり、他のプレイヤーよりも短時間で到達可能な領域を意味する。

これらの文献における優勢領域は、フィールドスポーツの画像を量子化したものに対して、ボロノイ領域(付録1を参照)における「最短距離」の要因を、「最短到達時間」に拡張したものである。従って各プレイヤーに対して、位置、速度、加速度ベクトルを設定することで、単位時間内に到達できる範囲を求めるものである。

ここで優勢領域は画像情報を入力とし、全ての画素に対する走査処理を要する。しかし、DVE では共有オブジェクトの位置情報から解析的に、管理権を所有している peer が容易に判別し得るべきである。これが実装されれば、位置情報のみの入力からリアルタイムでの処理が可能となる。そこで、仮想空間でのアバタの動きに基づいた勢力範囲の概念を AtoZ として提案する。まず、各アバタが一定時間内に到達できる範囲を考える。ここで、遠隔 peer において、次の離散時刻でのアバタの存在確率は、属性の時間的連続性を考慮して、正規分布で近似できると考えられる[13]。

そこで、2次元空間を例とすれば、図3で時刻 $T=t_0$ においてアバター A が上方に進行しているとき、現時刻 $T=t_0$ での位置と次の離散時刻 $T=t_1$ での位置を焦点とした楕円をアバター A の離散時刻分の進行範囲と見なすことができる。この進行範囲、すなわち 1 時刻後の存在確率が2次元正規分布に従っていると考えると、二次判別分析モデルでよく知られるマハラノビスの距離[15]をこの範囲の計算に利用できると思われる(付録2を参照)。

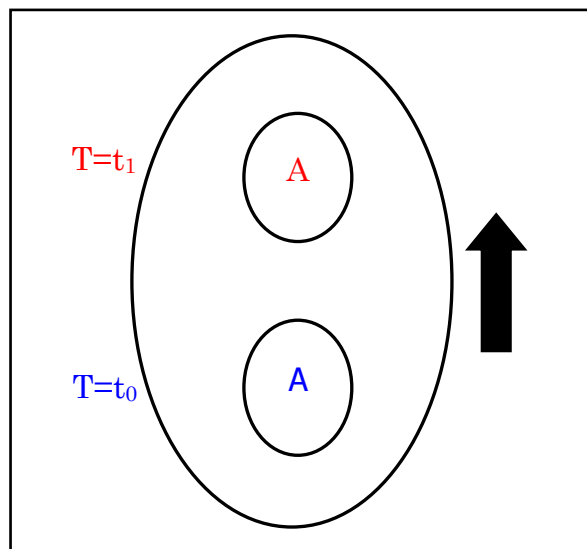


図3. アバター A の進行範囲
Fig.3 The marching range of avatar A

マハラノビスの距離は、次式で定義される。

$$D^2 = \frac{z_1^2}{V(z_1)} + \frac{z_2^2}{V(z_2)} \quad (4)$$

この式の z_1 , z_2 は軸散布図上のある1点(z_1 , z_2)であり、 $V(z_1)$, $V(z_2)$ はその分散、 D^2 がマハラノビスの距離[13]となる。ここで、分散 $V(z_1)$, $V(z_2)$ は、式(4)の楕円方程式[12]における長軸、短軸に対応する(付録3を参照)。

また、離散時刻をフレームと見なせば、次フレームでの予測位置は、この確率分布において中心(楕円の中心)付近である確率が最も高く、逆に楕円の縁に到達する確率は最も低い。このことより、本論文では求める楕円の2焦点として、現時刻での位置と1フレーム後の予測位置の2点を採用する。これにより、0.5フレーム後の予測位置と確率分布の中心が一致する。

これらの議論から AtoZ の計算には、得られた進行範囲の方程式をもとに、マハラノビスの距離における判別を用いれば良い(図 4)。このとき、AtoZ の境界線は 2 次曲線となる。また、図 4 中の D_A^2 、 D_B^2 は、アバタ A、B からのマハラノビスの距離である。

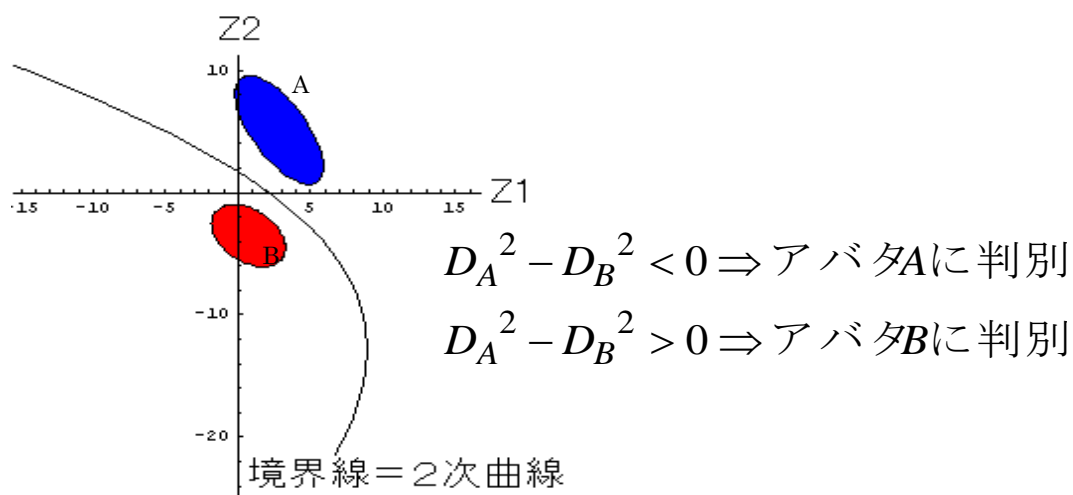


図 4. 判別の境界線
 Fig.4 The border line of discrimination

3. 2 相補予測の計算方法

本研究室では、P2P 型の DVE 上における通信遅延の影響を、予測により軽減する手法を提案している [6] [7]。この手法が適用された DVE では、各 peer 上で相手アバタの現在の状態を予測提示することが可能となる。ここで、Remote peer の予測アルゴリズムが既知である場合、Remote peer が予測する自己アバタの状態を提示することが可能となる [14]。

本論文では現在の両アバタの状態を予測するために、簡単な物理モデルを導入する。具体的には、短い時間間隔において、相手アバタの状態変化の加速度が変わらないことを仮定した力学的モデルである。図 5 は、peer A における両アバタの状態である。図 5 において、アバタ B の時刻 t_4 における現在の状態 $\underline{B}(t_4)$ は

$$\begin{aligned} \underline{B}(t_4) = & B(t'_3) \\ & + (B(t'_3) - B(t'_2))(t_4 - t'_3)(t'_3 - t'_1) / (t'_3 - t'_2)(t'_2 - t'_1) \\ & - (B(t'_2) - B(t'_1))(t_4 - t'_3)(t'_3 - t'_2) / (t'_2 - t'_1)^2 \end{aligned} \quad (5)$$

と書くことができる。またこの式を自己アバタにも適用することで、現時刻においてプレイヤーに対して Remote peer が予測する自己アバタの状態 $\underline{A}(t_4)$ を提示できる。

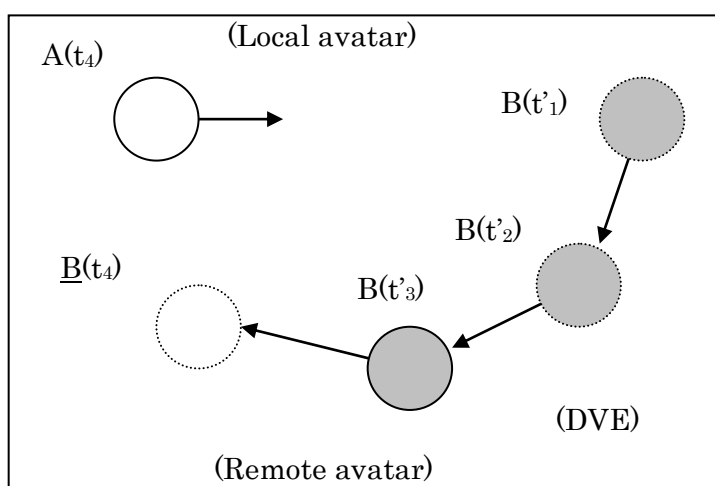


図 5. peer A における両アバタの状態
Fig.5 The state of both avatars
on the peer A

このとき Remote peer において、どの時点までの自己アバタの状態が知られているかを知る必要がある。図 6 は各 peer のタイムチャートである。ここで、peer A 側の時刻 $T=t_i$ において、peer B で受信されたアバタ A の最新状態の時刻を推測する。

ここで必要となる送信に要した時間 Δt_a は、あらかじめ NTP など同期を確保しておけば求められる。

そこで、求める時刻 t_n は次式で得られる。

$$t_n \leq t_i - \Delta t_a \quad (n = 0, 1, \dots, i) \quad (6)$$

この不等式を満たす最新の時刻 $T=t_n$ が、peer B で保有されるアバタ A の最新状態の時刻である。こうして得られた時刻 $T=t_n$ までの状態履歴を用いて自己アバタの予測を行うものとする。

このように peer 間で共通に保有されるデータのみを用い、同一の計算を行うことで、peer 間での計算結果が一致する。したがって、peer 間で予測提示の結果が合致し、情報提示の差異を抑制する。

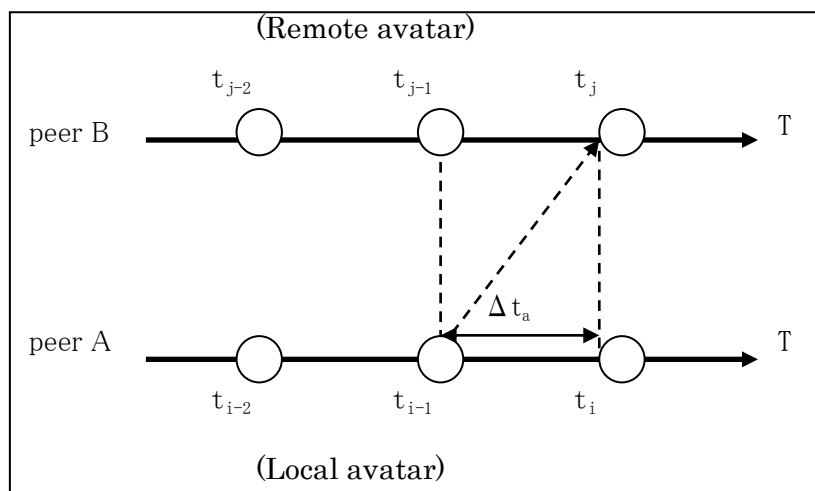


図 6. 各 peer のタイムチャート
Fig.6 The time-chart of each peer

第4章 クリティカル・ケース

本章では、通信遅延により発生する peer 間のクリティカル・ケースとその対処法について説明する。

4. 1 クリティカル・ケース

ここまでの段階で、AtoZ に基づいて管理者を決定することで共有オブジェクトの管理が可能となる。しかしながら、AtoZ の共通認識には、まだ予測不可能な事象に対する通信遅延分の曖昧さ（遅延による影響の軽減）の問題が残っている。つまり、共有オブジェクトに対して複数のアバタが同時にアクセスする際、通信遅延による微小なアバタ属性の差異が生じ、それにより AtoZ の計算に微妙な差異が発生することがある。こうしてオブジェクトの管理権に不整合が発生し、結果的に双方が管理権を取得したり、双方が委譲してしまうことがあり得る。このような際に、peer 間で発生事象に矛盾が起きることをクリティカル・ケースとよぶ。言い換えればクリティカル・ケースとは、通信遅延により各 peer での共有オブジェクトに対する干渉結果の一意性が破綻するケースである（図7）。

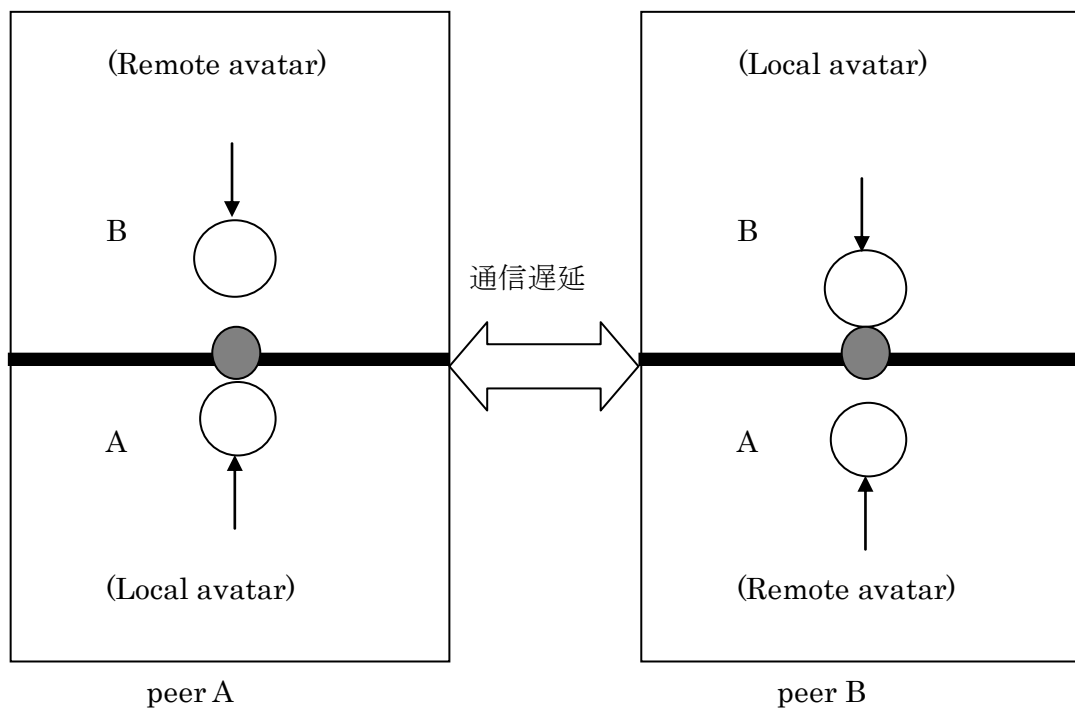


図7. クリティカル・ケースの概要
Fig.7 The Over-view of the critical-case

4. 2 Dead Zone

クリティカル・ケースの発生は、複数のアバタの AtoZ 境界付近への集中により起こる。このため AtoZ 境界付近でアバタの存在確率が拮抗する領域を、Dead zone とし、管理権の計算を厳密に行う。つまり、遅延とフレーム間隔によって生ずる情報損により管理権が不明確となる領域と考えて、Dead Zone は次式で定義できる。

$$DZ(A, B) = \left\{ x \mid x \in Z, \frac{|Time(O, A, x_A) - Time(O, B, x_B)|}{\Delta t} < ([d / \Delta t] + 1) \right\} \quad (7)$$

ただし、式中の [] はガウス記号となる。ここで、DZ(A, B) はアバタ A, B 間の Dead Zone、d[ms] は通信遅延、 Δt [ms] は更新間隔となっており、 $([d / \Delta t] + 1)$ で遅延フレーム数を算出できる。その他は式(1)と同様である。そこで、Dead Zone の幅は、両アバタの位置と最高速度、および通信遅延、更新間隔に依存していると考えられる。したがって、Remote peer にデータが届くまでに各アバタが最速で進行したとしても、クリティカル・ケースが発生しないような領域のみを AtoZ とし、どの AtoZ にも属さない領域を Dead Zone とする。すると、この幅は

$$Border = MaxSpeed \times ([d / \Delta t] + 1) \times Scale \quad (8)$$

となる。ここで、MaxSpeed[cm/frame] はアバタの最高速度、d, Δt は式(7)と同じである。また、Scale は仮想空間を実空間にスケーリングするための定数である。この式によって通信遅延の間に相手アバタが進める最大距離を求められる。したがって、ここで得られた Border の値をマハラノビスの距離における判別の閾値として用いることで、適切な Dead Zone の幅を算出できると考えられる。

Dead Zone 内に共有オブジェクトがある場合、式(4)で管理者を決めることはできない。また、管理権の移行が頻繁に行われるこのようなケースで、これを円滑に行うには、メッセージ通信なしに決定する必要がある。更に、管理者以外の peer で共有オブジェクトへのアクセスがあった場合も応答性の低下が起こることから、Dead Zone 内の管理者は、次に共有オブジェクトへアクセスする確率が最も高い peer にすべきである。このことより、本論文では図8に示す状態遷移図を用いて全 peer が Dead Zone 内の管理者を各々独自に決定する。ここで、共有オブジェクトが Dead Zone 内に入った瞬間、Neutral 状態（管理権が明白でない状態）に遷移する。この状態で Dead Zone 内の管理者を図中の不等式により決定する。ここで、 $A(t)$, $B(t)$, $0(t)$ は、各 peer が保有する時刻 t におけるアバタ A、B および共有オブジェクト 0 の位置情報である。

これは AtoZ の計算と類似しており、管理権に不整合が発生し難い。また、仮に発生しても、Dead Zone 進入時には毎回再計算され、収束し易いため不整合の連続発生を抑制できる。したがって合理的なクリティカル・ケースへの対応であると考えられる。

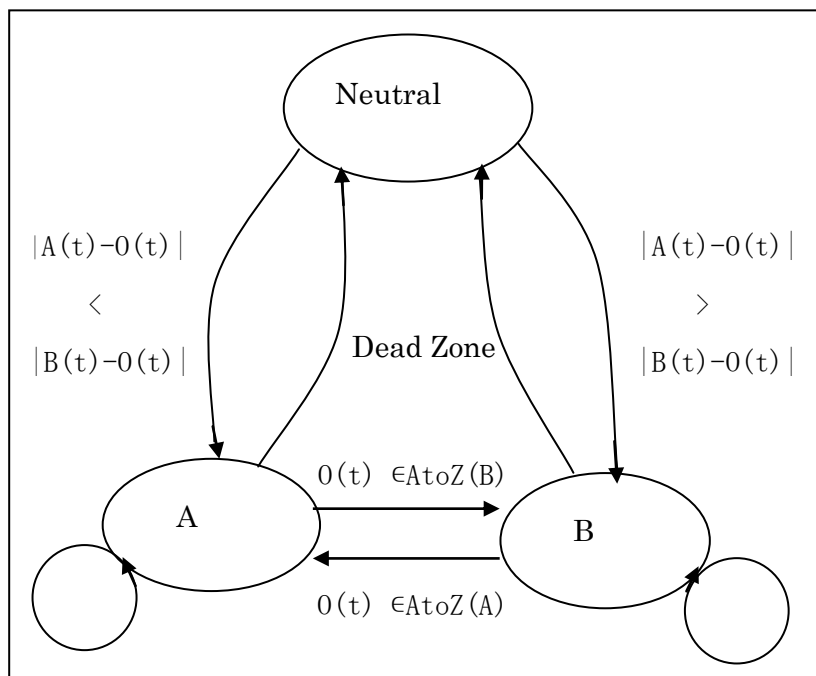


図8. 管理者の状態遷移図
Fig.8 The state transition diagram for the determination of the owner

第5章 実験

今回、提案した手法の有効性と適用範囲について評価するため、システム評価及び被験者実験を行った。

5.1 実験タスク

以上の概念を導入したアプリケーション例としてネットワーク対戦型エアホッケーを試作した(図9)。この図において左側は実際のゲーム画面、右側はAtoZのようすを表した画面である。ここで、白色パックが共有オブジェクトであり、パックを打つ白黒のマレットが各アバタとなる。また、AtoZ画面では、白円、上方楕円、下方楕円がそれぞれ、パック、相手マレット、自己マレットを表す。また、中央の黒い帯がDead Zone、その下側領域が自己マレットの、上側領域が相手マレットのAtoZを表す。ここで、試作したエアホッケーではフィールド内を互いに自由に移動可能とした。これは、フィールド型DVE全般におけるAtoZ、Dead Zone、クリティカル・ケースの評価を行うためである。

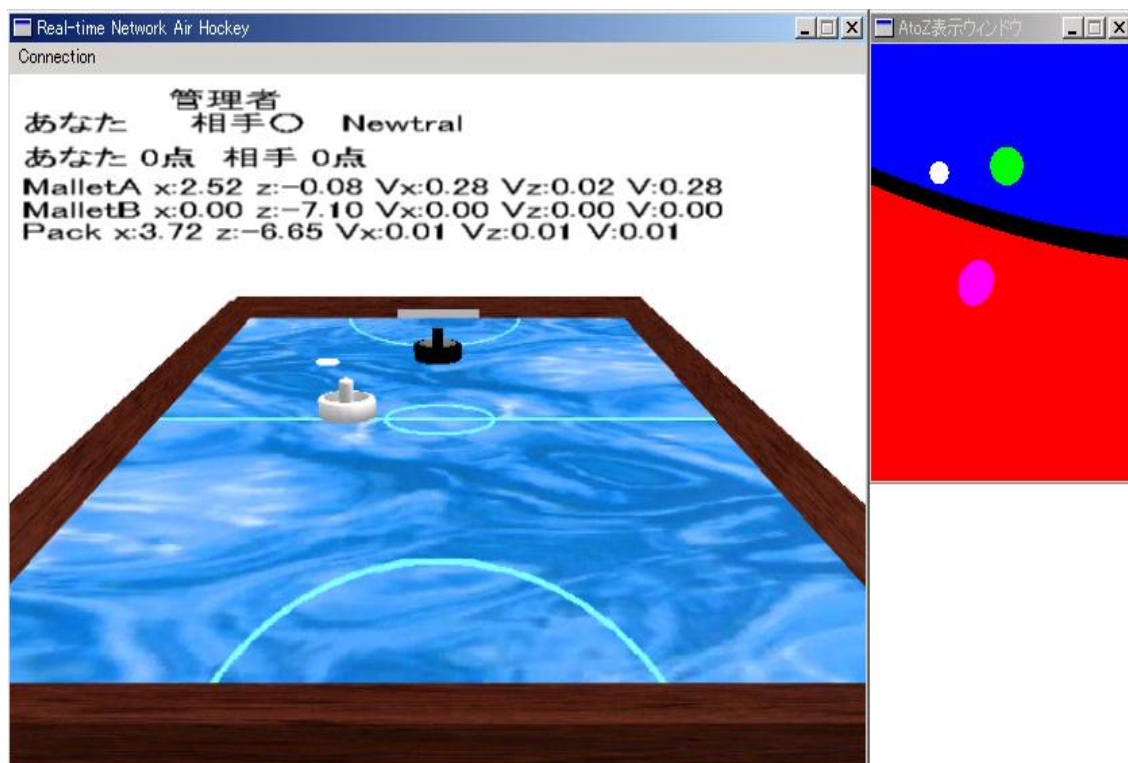


図9. エアホッケーのゲーム画面

Fig.9 The game-screen of the Air Hockey

また、ネットワーク対戦型エアホッケーにおける仮想空間、およびシステムの設定条件を次に示す。

仮想空間

- ・対戦台の大きさ：縦 2.89[m]，横 1.48[m]
- ・台の動摩擦係数： 3.0×10^{-5}
- ・パック半径，マレット半径：4.1[cm]，8.5[cm]
- ・パック速度，マレット速度の上限：3.75[m/sec]

システム

- ・同期取得のための時刻情報の通信：TCP
- ・ゲーム中の情報通信：UDP
- ・更新間隔：20[ms]

5. 2 実験の目的

この実験における評価の対象は

- ①共有オブジェクトの一貫性が保てるか
- ②共通の管理者を全 peer で認識できるか
- ③全 peer で全アバタの物理属性が一致するか

となる。まず、①はゲーム性を維持するために最優先されなければならない。①を満たすためには、②が満たされている必要がある。②は AtoZ によって決定され、AtoZ の一貫性は③によって保証される。

したがって、これらを満たすための評価項目は

- i) peer 間でのマレット位置の差異
- ii) peer 間での管理権が一致する確率
- iii) peer 末間でのパック位置の差異

となる。以上の項目を実験によって評価する。ここで模擬遅延は文献[16]で提供されている NistNet を用いて発生した。遅延モデルは遅延なし(Type 0)、イントラネットを想定した小規模遅延(Type 1)、ならびにインターネットを想定した大規模遅延(Type 2)の3種類を用いた。

5. 3 実験方法と結果

実験は 20 代の被験者 12 人 10 ペアで行った。ここで、実験条件は次のとおりである。

実験条件

- ・ 予測方法：予測なし、予測、相補予測
- ・ 遅延の種類：Type 0：遅延なし，Type 1：平均 8.7[ms]，標準偏差 4.1[ms]，Type 2：平均 57.0[ms]，標準偏差 8.1[ms]

この条件で、各被験者ペアに対してエアホッケーのゲームをしてもらい、実験を行った。ここで相補予測では、Remote peer から見た Local peer の状態も合わせて提示するようにした。実験は、予測方法（3種類）と遅延の種類（3種類）の組み合わせで 9 試行を行い、1 試行は 2 点先取とした。実験結果の一例を図 10-24 に示す。ここで、図 10-18 は peer 間のマレット位置の差異であり、図 19-24 は peer 間のパック位置の差異を示している。また、それらをまとめた結果を表 1-4 に示す。これらの表中の上段は、それぞれの試行における平均を示しており、下段の()内は標準偏差を表す。

また補足として、同時に行ったアンケート結果を図 25-27 に示す。アンケート項目は、不公平感（得点情報に関する認識の不一致）、操作性、楽しめたかの 2 項目である。図中の横軸の番号は、「遅延の種類－予測方法（予測なし：0，予測：1，相補予測：2）の組み合わせ」を示す。ここで、不公平感について説明する。今回、試作したエアホッケーゲームでは、得点情報は得点された方が相手に通知するプロトコルとなっている。このとき、通信遅延によって Local peer では得点したと以为っていても、Remote peer では得点されていないと判断されることがあり得る。これによって、得点情報に関する認識の不一致が生じる。このようなケースが生じたことを、プレイヤーが感じられたかについてアンケート調査を行ったものである。

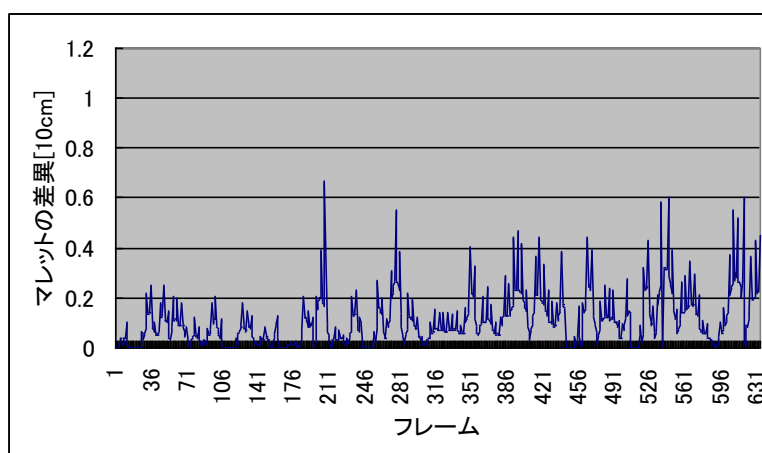


図10. Peer 間のマレット位置の差異(Type 0, 予測なし)
 Fig.10 The difference of mallet position consistency(Type 0, without prediction)

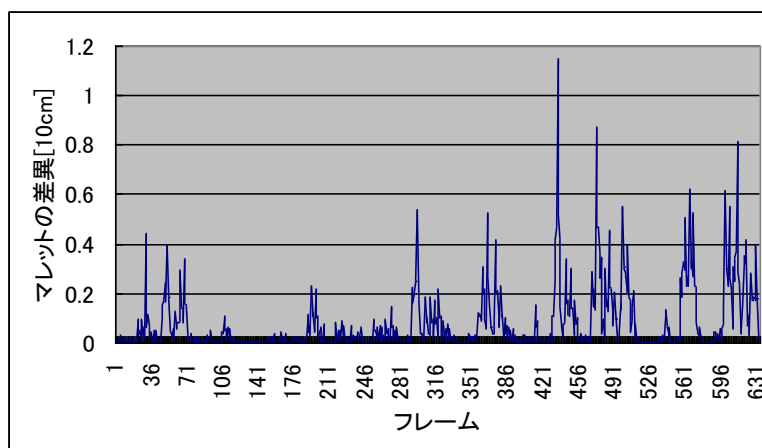


図11. Peer 間のマレット位置の差異(Type 0, 予測)
 Fig.11 The difference of mallet position consistency(Type 0, prediction)

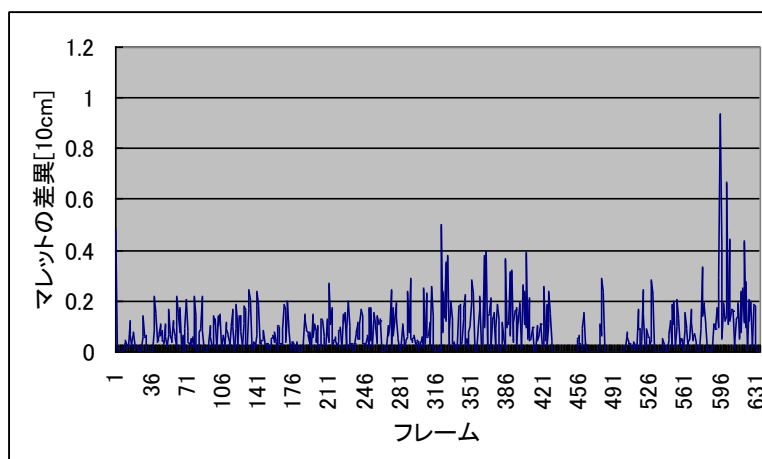


図12. Peer 間のマレット位置の差異(Type 0, 相補予測)
 Fig.12 The difference of mallet position consistency(Type 0, mutual prediction)

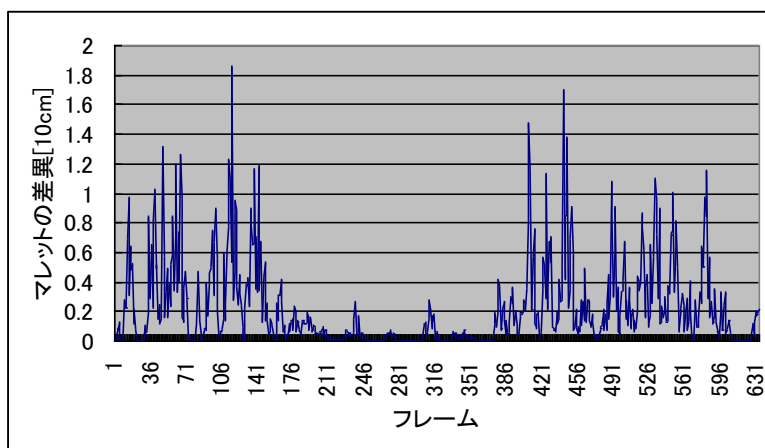


図13. Peer 間のマレット位置の差異(Type 1, 予測なし)
 Fig.13 The difference of mallet position consistency(Type 1, without prediction)

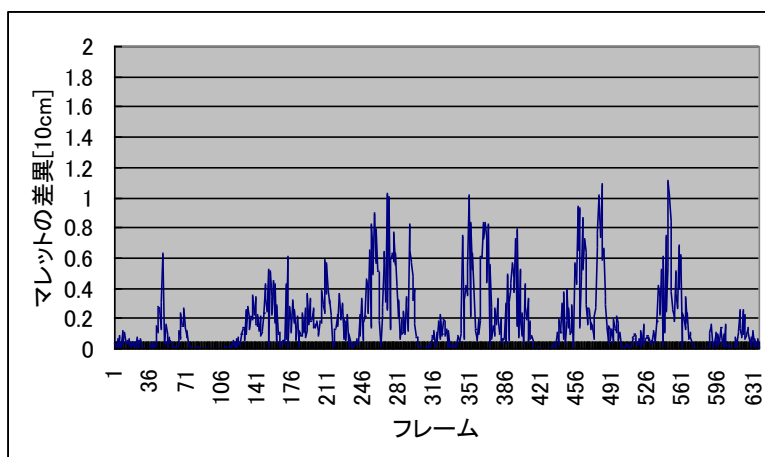


図14. Peer 間のマレット位置の差異(Type 1, 予測)
 Fig.14 The difference of mallet position consistency(Type 1, prediction)

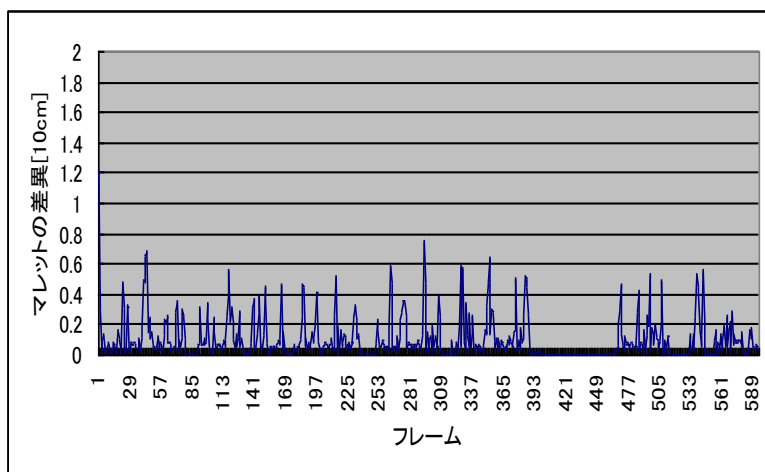


図15. Peer 間のマレット位置の差異(Type 1, 相補予測)
 Fig.15 The difference of mallet position consistency(Type 1, mutual prediction)

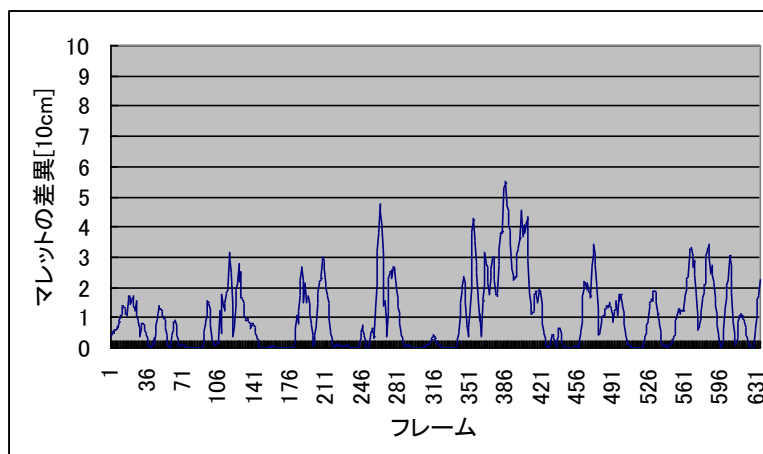


図16. Peer 間のマレット位置の差異(Type 2, 予測なし)
 Fig.16 The difference of mallet position consistency(Type 2, without prediction)

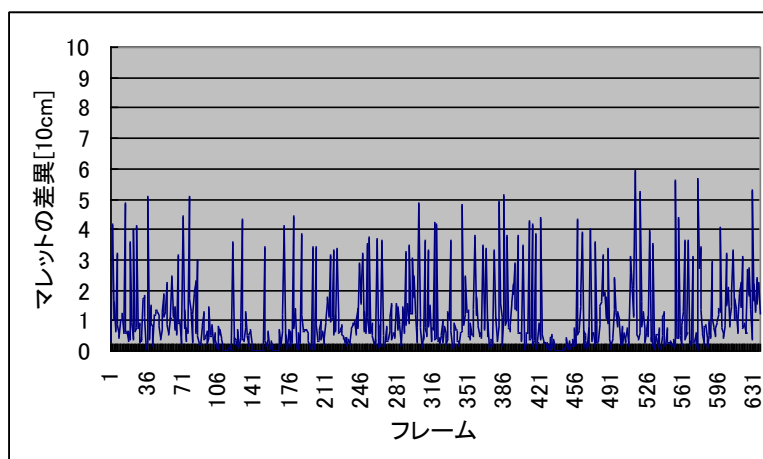


図17. Peer 間のマレット位置の差異(Type 2, 予測)
 Fig.17 The difference of mallet position consistency(Type 2, prediction)

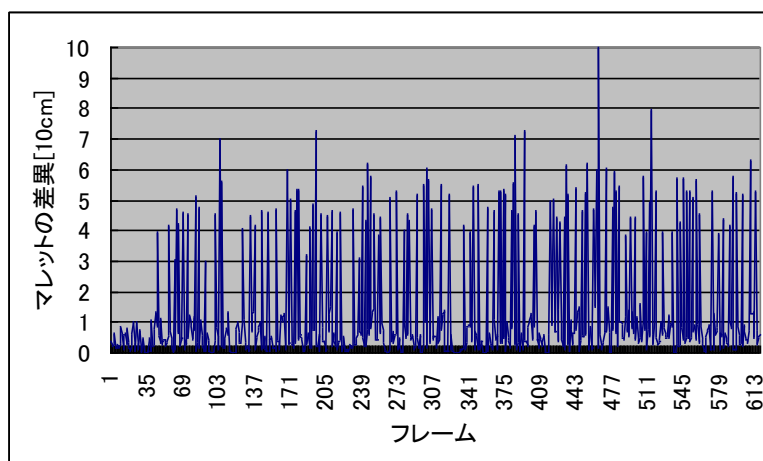


図18. Peer 間のマレット位置の差異(Type 2, 相補予測)
 Fig.18 The difference of mallet position consistency(Type 2, mutual prediction)

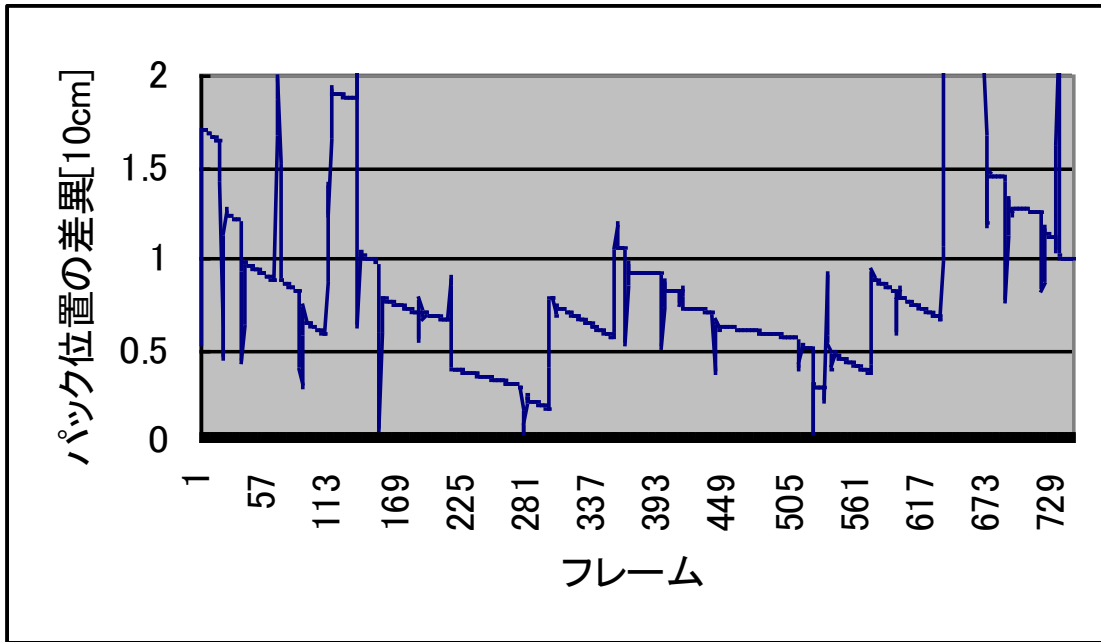


図19. Peer 間のパック位置の差異(Type 0, 予測なし)
 Fig.19 The difference of pack position consistency(Type 0, without prediction)

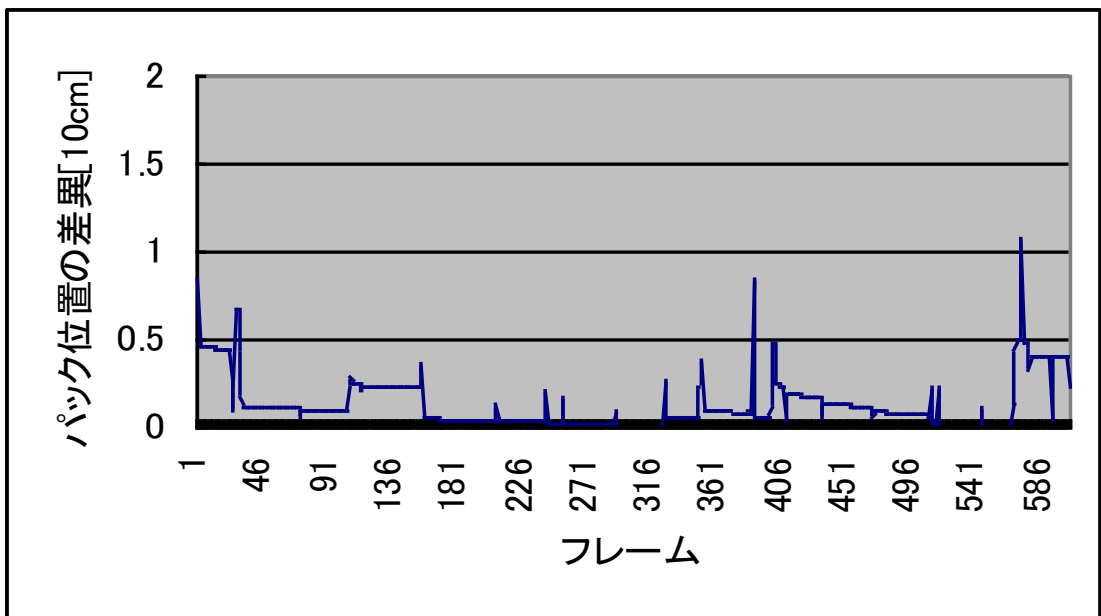


図 20. Peer 間のパック位置の差異(Type 0, 予測あり)
 Fig.20 The difference of pack position consistency(Type 0, with prediction)

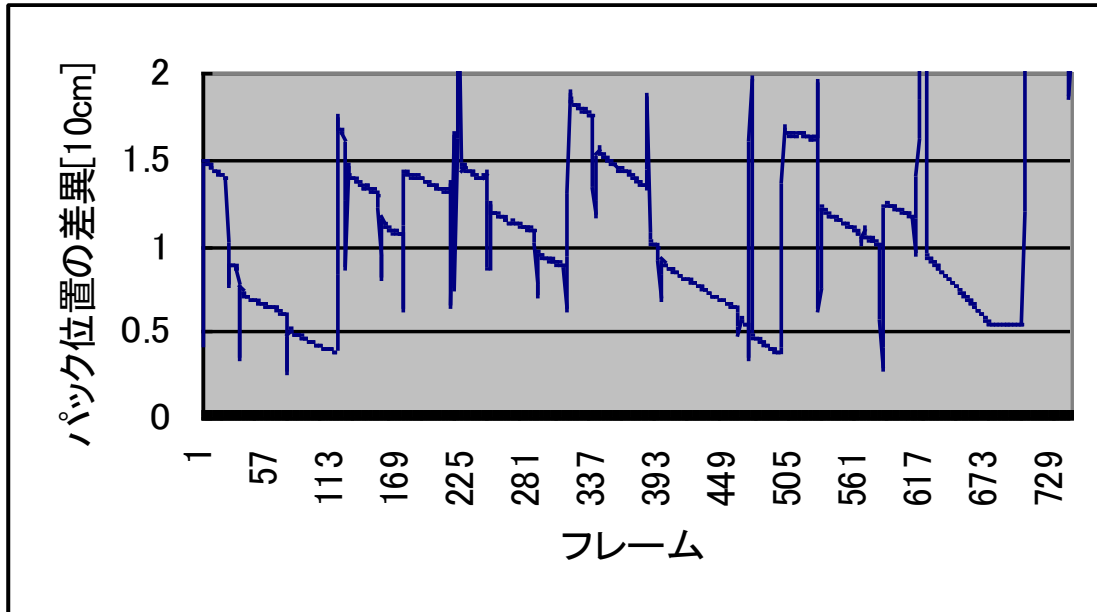


図 21. Peer 間のパック位置の差異(Type 1, 予測なし)
 Fig.21 The difference of pack position consistency(Type 1, without prediction)

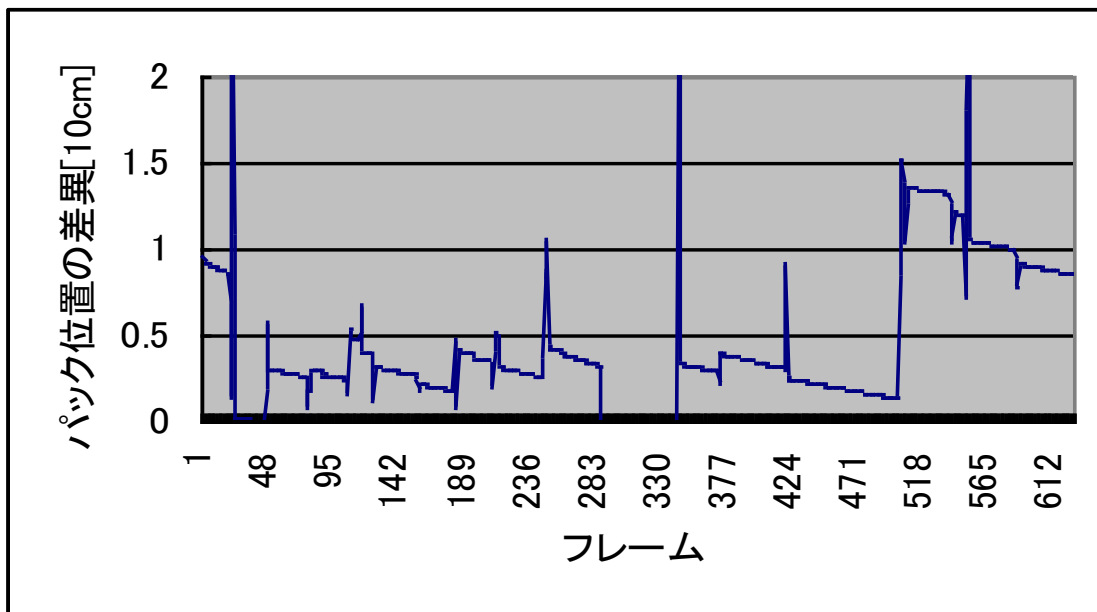


図 22. Peer 間のパック位置の差異(Type 1, 予測あり)
 Fig.22 The difference of pack position consistency(Type 1, with prediction)

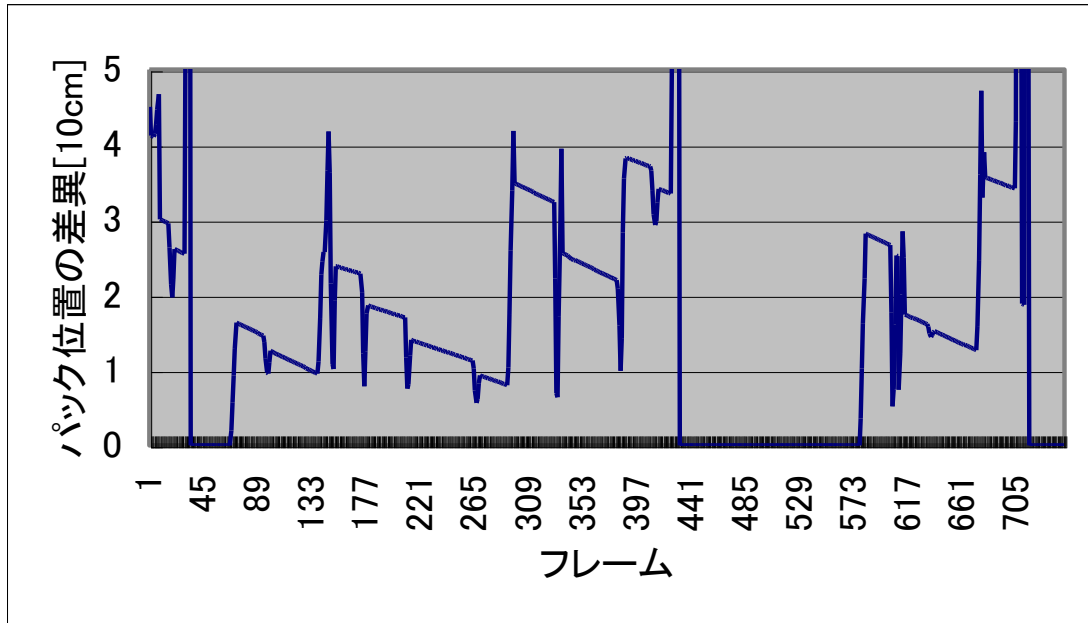


図 23. Peer 間のパック位置の差異(Type 2, 予測なし)
 Fig.23 The difference of pack position consistency(Type 2, without prediction)

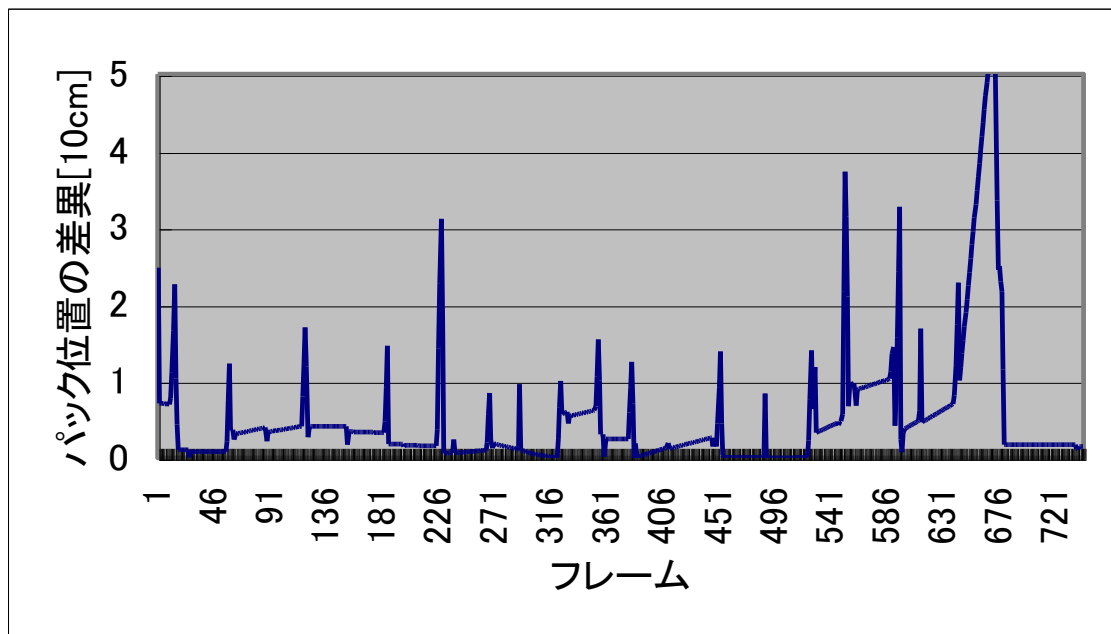


図 24. Peer 間のパック位置の差異(Type 2, 予測あり)
 Fig.24 The difference of pack position consistency(Type 2, with prediction)

表1. Peer 間のマレット位置の差異
 Tab.1 The difference of mallet position consistency (単位:10cm)

	Type 0	Type 1	Type 2
予測なし	0.140531	0.242195	0.80978
	(0.03450)	(0.15391)	(0.14926)
予測	0.077826	0.177317	0.775951
	(0.02095)	(0.04729)	(0.35485)
相補予測	0.078876	0.133286	1.001106
	(0.04505)	(0.10814)	(0.19429)

表 2. Peer 間のパック位置の差異
 Tab.2 The difference of pack position consistency (単位:10cm)

	Type 0	Type 1	Type 2
予測なし	1.047164	1.185152	2.161932
	(0.17331)	(0.30476)	(0.31743)
予測	0.783663	0.690508	0.766295
	(0.03778)	(0.10802)	(0.18367)
相補予測	0.834139	0.829484	0.760241
	(0.15660)	(0.23246)	(0.14222)

表 3. ゲーム全体で管理者が一致する確率
 Tab.3 The percentage of ownership consistency (単位 : %)

	Type 0	Type 1	Type 2
予測なし	94.3	94.5	89.5
	(2.32594)	(2.37697)	(6.15224)
予測	96.1	95	92.7
	(2.25610)	(3.37639)	(3.74299)
相補予測	95	94.5	93.22222
	(3.46410)	(2.33452)	(3.15446)

表 4. Dead Zone 内で管理者が一致する確率
 Tab.4 The percentage of ownership consistency in the Dead Zone (単位 : %)

	Type 0	Type 1	Type 2
予測なし	74.9	67.7	72.3
	(7.244998)	(6.82715)	(9.39202)
予測	77.1	78.3	78.4
	(8.88200)	(10.32521)	(7.68375)
相補予測	78.55556	73.1	80.66667
	(9.776515)	(3.04795)	(9.07989)

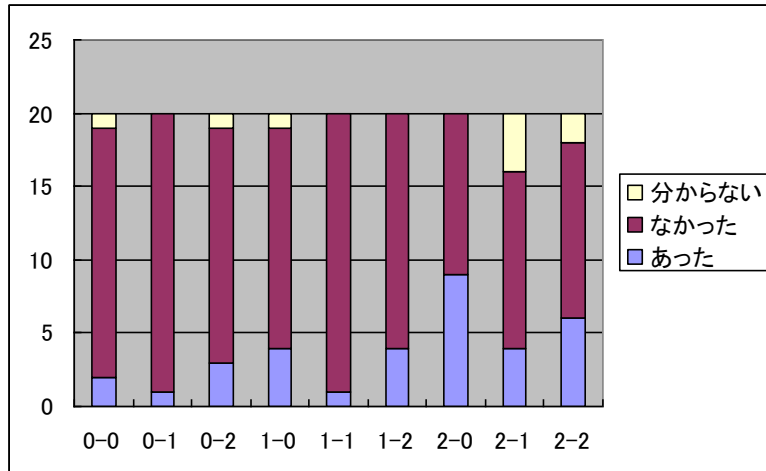


図 25. 不公平感
Fig.25 The sense of unfairness

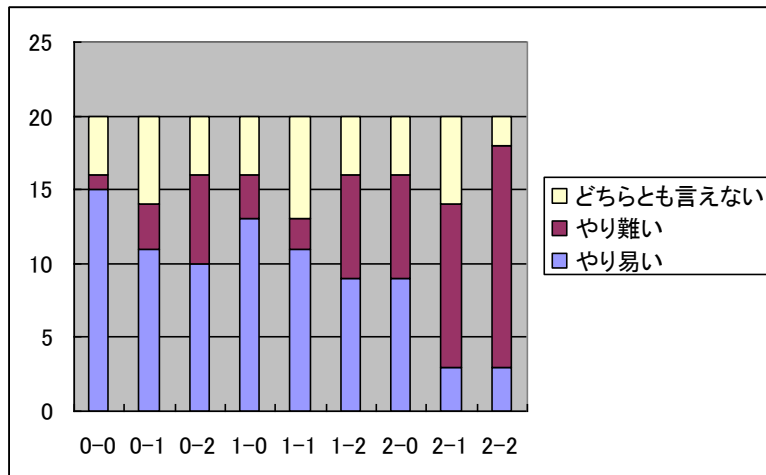


図 26. 操作性
Fig.26 The operability of the mallet

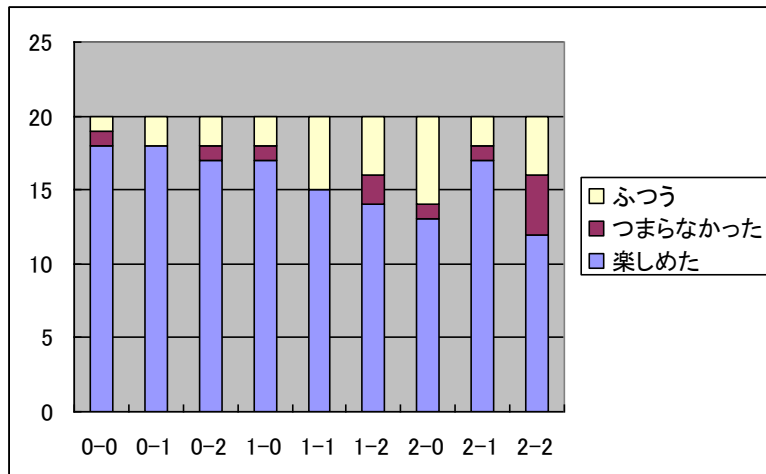


図 27. 楽しめたか
Fig.27 The sense of fun

5. 4 考察

表 1 および図 10-18 より、peer 間のマレット位置の差異は、遅延量が少なければ、予測なし、予測、相補予測の順で軽減されている。しかしながら、マレットの動きは一般に激しいため、遅延量が大きい場合、加速度一定の仮定は成り立たず、簡単な物理モデルでは予測が困難になると考えられる。特に、相補予測を行った場合では、相手アバタだけでなく自己アバタに対しても予測を行うため、予測が外れたときの誤差が大きくなる。以上より、マレットの動きを予測する上での遅延量の限界は、Type 1 と Type 2 の間に存在すると考えられる。また、予測なしと予測(相補予測)を比較すると、予測なしは誤差が連続的であるのに対し、予測(相補予測)は誤差が小刻みに揺れている。その結果、予測(相補予測)を行った方が平均値は低かったとしても誤差の幅は大きくなってしまふ。これは、予測誤差を急激に引き戻しているためである。そのため、プレイヤーからはマレットがぶれて見えてしまふことになり、操作性の劣化を生じる(図 26)。

表 2 および図 19-24 より、予測なしより予測、相補予測を行った方が、明らかにパック位置の差異は軽減されている。ここで、パックは共有オブジェクトであるため、予測、相補予測ともに共通の予測方法であり、遅延の種類に関わらず、常にパック 1 個分程度のずれが生じているが、予測なしでは遅延に比例したずれとなっている。これは、パックの動きが摩擦を考慮した滑らかな等加速度運動と仮定できるためであると考えられる。

表 3 より、いずれのケースにおいても概ね 90%以上の確率で管理者の一致が見られる。ただし、予測なしで遅延大のときに管理者一致率が 90%を割っている。逆に、表 4 では微小なずれの影響が顕著に見られる。予測なしより、予測または、相補予測の方が、管理者一致率は高かった。この結果と表 1, 2 を総合すると、マレットのずれ(つまり AtoZ のずれ)よりもパックの位置ずれの方が管理者の不一致を引き起こしやすいと思われる。式(7)によりマレット間の距離の増加に伴い、Dead Zone の幅は狭くなる。このような状況でパック位置のずれが発生すると、パックの Dead Zone への進入と退出のタイミングが peer 間で大きくずれてしまひ、両 peer で管理者の一致する時間が短くなる。ただし、このケースではアバタ同士が離れているため、クリティカル・ケースにはつながり難い。

クリティカル・ケースは、全試行中で合計4回発生した。内訳は、Type 0の予測、相補予測、Type 1の相補予測、Type 2の相補予測がそれぞれ1回である。予測なしでは、クリティカル・ケースの発生は確認できなかった。これより、Dead Zone 付近では、予測が外れた場合に双方が管理権を取得する確率が高くなってしまい、クリティカル・ケースが発生しやすくなると考えられる。特に、相補予測ではその危険性が高くなるといえる。ここで、予測方法毎のクリティカル・ケース発生頻度を求めると、分母となる総フレーム数約9万フレームのうち、予測なし：0、予測：1、相補予測：3(フレーム)となる。以上より、Dead Zone 付近では予測なしの方が安全であると考えられる。

図 25 より、不公平感は2-0のケースで最も多く感じられたという結果になった。これは、遅延大のときにパックの予測を行わないために、パック位置の認識に peer 間で大きなずれが生じてしまい、得点が入らないように感じられたと考えられる。

図 26 より、操作性については遅延が大きいほど、また予測を行うほど、その劣化が見られた。これは、予測を行うことでマレットがぶれて見えてしまうためである。特に、相補予測では相手マレットだけでなく、Remote peer から見た自己マレットについても予測を行うため、それがぶれてしまうと非常にプレイし難くなると考えられる。

図 27 より、遅延大のケース以外では、ゲームとしても十分機能していることが知られた。

第6章 まとめ

本論文では、P2P のフィールド型 DVE を構成する際に問題となる共有オブジェクトの管理権、ならびに各 peer での物理属性の一貫性を保つためのプロトコルを提案した。その結果、AtoZ、相補予測、クリティカル・ケースへの対処における有効性が確認できた。これらのプロトコルは、P2P のフィールド型 DVE 全般に適用できると考えられる。

今後は、多人数参加型あるいは複数の共有オブジェクトが存在するフィールド型 DVE において、本手法が有効であるか、それらを実際に構築することで確認する予定である。また、アンケート結果から予測を行った場合、プレイヤーからマレットがぶれているように見え、操作し難いという意見が得られた。そこで、アバタ動作に対する適切な予測モデルと予測誤差の convergence（予測の収束）が課題となる。

謝辞

本研究に関して多くの助言をいただき、さらに休む暇もなく仕事を与えて下さいました米倉達広助教授に深く感謝申し上げます。そのおかげで、やり甲斐のある素晴らしい研究と出会う事ができました。

また、些細な質問にも快く答えていただき、様々な議論をして下さいました埴大さん、大部由香さんに心より感謝致します。そして、本研究に関して数多くのご指摘をいただきました研究室の皆さん、実験などで本研究に携わったすべての方々に多大なる感謝の意を表します。

参考文献

- [1] 竹村治雄：“ネットワークを使ったコミュニケーション”日本バーチャルリアリティ学会誌, vol.4,no.1,pp.59-63,1999.
- [2] 橋本孝之, T.B.Sheridan,M.V.Noyes：“時間遅れを有するテレオペレーションにおける予告情報の効果,”人間工学, vol.22,no.2,pp.91-92,Mar.1986.
- [3] 波多野健, 山本泰秀, 高松亮, 佐藤誠：“予測動作提示による仮想遠隔共同作業環境の時間遅れ補償,”日本バーチャルリアリティ学会大会論文集, vol.1, pp.155-158, 1996.
- [4] 河野義広, 埴大, 米倉達広：“相補予測機能を用いたネットワーク対戦型エアホッケーの試作”, 日本バーチャルリアリティ学会第8回大会, pp383-386, 2003.9.
- [5] 河野義広, 埴大, 米倉達広：“相補予測機能を用いた P2P 型ネットワーク対戦エアホッケーの試作”, 電子情報通信学会, MVE2003-68, pp17-22, 2003.10.
- [6] 米倉達広, 埴大：“遅延を伴う分散仮想環境での状態予測を用いた協調型タスクのパフォーマンス劣化補償”, 日本バーチャルリアリティ学会論文誌, Vol.7, No.4, pp.543-554, 2002.12.
- [7] 米倉達広, 埴大：“遅延のあるネットワークを介した仮想空間上の複数アバタのための適応型動作予測フィルタ”, 電子情報通信学会論文誌 A 分冊.
- [8] 川原圭博, 松本延孝, 森川博之, 青山友紀：“ピアツーピアネットワーク型仮想環境における更新情報共有手法”, 信学総大, 2002.3.
- [9] 川原圭博, 松本延孝, 森川博之, 青山友紀：“ネットワーク仮想環境のための分散型通信アーキテクチャ”, 電子情報通信学会技術研究報告, IN2001-229, March 2002.
- [10] 瀧剛志, 長谷川純一：“勢力範囲に基づいたチームスポーツ解析”, 情報処理, 42 巻 6 号, pp.582-586, 2001.
- [11] 瀧剛志, 長谷川純一：“チームスポーツにおける集団行動解析のための特徴量とその応用”, 電子情報通信学会論文 D-II Vol.J81 D-II No8 pp.1802-1811, 1999.
- [12] S.K. Singhal, M.R. Zyda: Networked Virtual Environments: Design and Implementation, Addison-Wesley, Massachusetts,1999.
- [13] S.K. Singhal：“Effective remote modeling in large-scale distributed simulation and visualization environments”, Ph.D dissertation, Department of Computer Science, Stanford University, Palo Alto, August 1996.

- [14] I.N.ブロンシュテイン, K. A. セメンジャーエフ : “数学ハンドブック”, 森北出版株式会社, 1985.
- [15] 有馬哲, 石村貞夫 : “多変量解析のはなし”, 東京図書, 1987.
- [16] NISTnet,<http://snad.ncsl.nist.gov/itg/nistnet/>
- [17] 笹本綾子, 米倉達広 : “遅延を予測したリアルタイムネットワーク対戦ゲームの試作”
平成 11 年度茨城大学工学部情報工学科卒業論文
- [18] 笹本綾子, 米倉達広 : “適応型誤差修正予測法によるリアルタイムネットワーク対戦ゲームの試作” 平成 13 年度茨城大学大学院理工学研究科情報工学専攻修士論文
- [19] 近藤俊哉, 米倉達広 : “ネットワーク対戦型システムにおけるユーザの振る舞い解析に関する研究” 平成 14 年度茨城大学大学院理工学研究科情報工学専攻修士論文
- [20] 大山崇のボロノイ図のページ <http://www.nirarebakun.com/voro/voro.html>
- [21] ゲームのためのやさしい数学
<http://www.geocities.co.jp/Playtown/7190/math/t02t1mC.html>
- [22] 情報統計学Ⅱ
<http://kuva.mis.hiroshima-u.ac.jp/~asano/Kougi/99s/JouhouToukei2/6-16.html>
- [23] 早川啄哉, 田口剛史, 中野美紗子, 辻 直樹, 谷川 敬一郎, 中村 健二, 竹内 克明 :
“DirectX8&VC++ 3D の基礎とゲームの作り方”, 工学社, 2002.

付録

1. ボロノイ領域

ボロノイ領域とは、平面上の各点を垂直二等分線によって分割した領域で、 $P = \{p_1, p_2, \dots, p_n\}$ をユークリッド空間中での n 個の点の集合、 $In = \{1, 2, \dots, n\}$ を n 個の整数の集合、 \mathbb{R}^2 を実数の集合としたとき、 $p_k \in P$ のボロノイ領域は次のように定義される。

$$V(p_k) = \{x \in \mathbb{R}^2 \mid d(x, p_k) \leq d(x, p_m) \text{ for } m \neq k, m \in In\} \quad (9)$$

ここで、 p_k はその点のラベルとその位置ベクトルの両方を示すものとし、 $d(x, y)$ は位置 y から x までのユークリッド距離を示す。また、 P に対するボロノイ領域の和集合をボロノイ図と呼ぶ。図 28 はボロノイ図の具体例である。

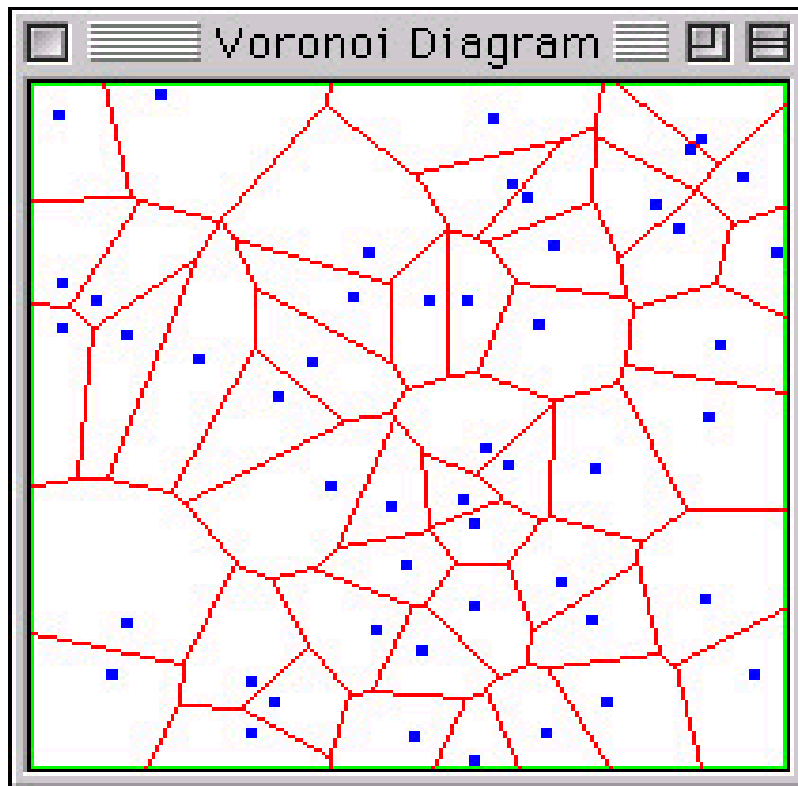


図 28. ボロノイ図の例
Fig.28 The example of voronoi daiagram

2. マハラノビスの距離

図 29 において散布図上に2つの変数 x_1, x_2 によるデータがあり、それが A, B どちらかのグループに分かれているとする。ここで、新しいデータ X がどちらに入るかを考える。普通に考えると、両グループの中心とのユークリッド距離を調べ、近い方に分類すればよい。このとき、両グループの境界は互いの中心の垂直二等分線によって分割できる(図 30)。

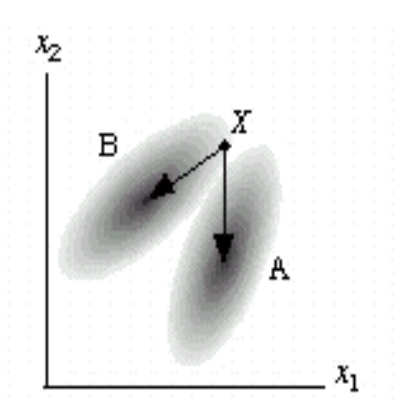


図 29. X は A, B どちらに近いか?
Fig.29 Which is X closer to A or B?

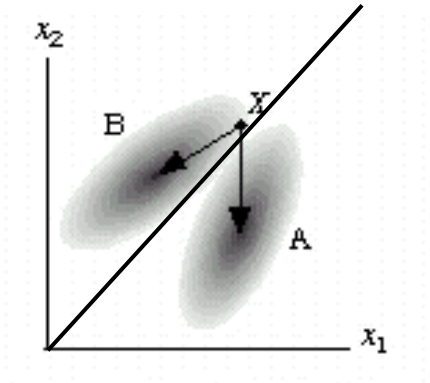


図 30. 線形判別関数
Fig.30 The linear discriminant function

この線形判別は、ボロノイ領域と一致する。しかしながら、図 31 のように両グループの分散が大きく違う場合は、これでは不十分である。そこで、通常のユークリッド距離ではなく、確率（密度）を考慮に入れた距離で計れるとよい。このような距離を「マハラノビスの距離」と言う。マハラノビスの距離は、ユークリッド距離を分散で標準化することで定義できる。

まず、あるグループの確率分布が、図 32 のように表されているとする。また、変量 x_1 , x_2 の平均をそれぞれ μ_1 , μ_2 , 分散をそれぞれ σ_1^2 , σ_2^2 とし、 x_1 , x_2 の相関係数を ρ とする。ここで、 x_1 , x_2 を式(10)のように標準化する。平均は u_1 , u_2 とも 0、分散はともに 1、相関係数は同様に ρ となる。また式(11)で、 u_1 , u_2 を主成分の変量 z_1 , z_2 に変換する (図 33)。

$$u_1 = \frac{x_1 - \mu_1}{\sigma_1}, \quad u_2 = \frac{x_2 - \mu_2}{\sigma_2} \quad (10) \quad z_1 = \frac{u_1 + u_2}{\sqrt{2}}, \quad z_2 = \frac{u_1 - u_2}{\sqrt{2}} \quad (11)$$

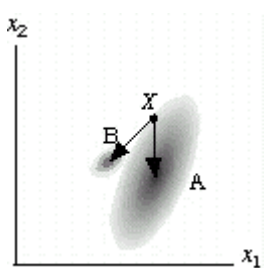


図 31. X は A に近い
Fig.31 X is near by A.

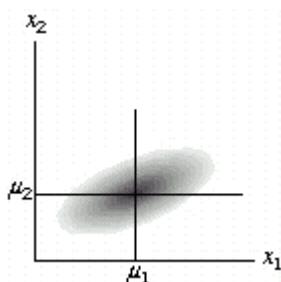


図 32. 2次元確率分布
Fig.32 Two dimension probability distribution

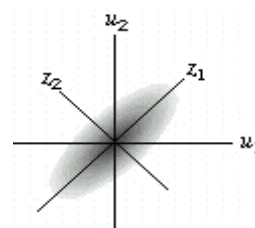


図 33. 主成分に変換
Fig.33 The transformation of the principal component

そして、散布図上のある1点 (z_1, z_2) と分布の中心 (x_1, x_2) 軸上では (μ_1, μ_2) , z_1, z_2 軸上では $(0, 0)$ との「分散で標準化した」距離(平方距離)を、三平方の定理より定義する。

$$D^2 = \frac{z_1^2}{V(z_1)} + \frac{z_2^2}{V(z_2)} \quad (12)$$

このように、 z_1, z_2 軸上でのそれぞれの分散 $V(z_1), V(z_2)$ で標準化した平方距離の和をマハラノビスの距離と言う。境界の判別は、グループA, Bのマハラノビスの距離を D_A^2, D_B^2 とすると、次の条件で判別できる。境界線は、 (x_1, x_2) の2次式 $=0$ となるため、散布図上の2次曲線となる(図34)。

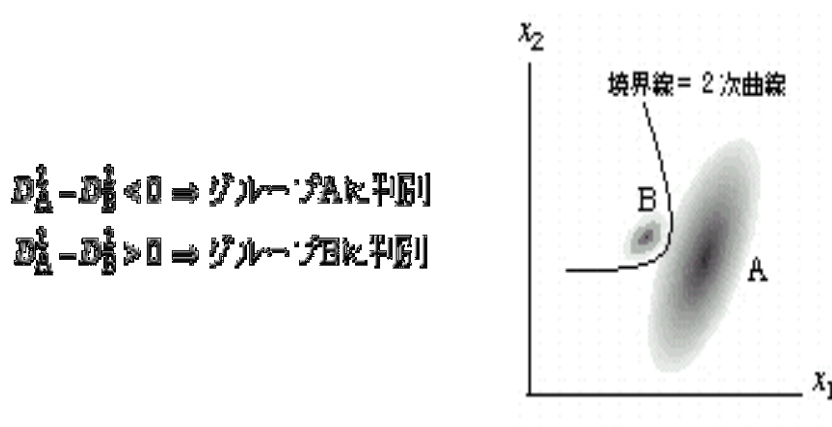


図 34. マハラノビスの距離の判別
Fig.34 The discrimination of Mahalanobis generalized distance

3. 2焦点から求める楕円方程式

まず、高校で習う楕円の方程式は、式(13)である。

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (13)$$

これは、原点を中心とし、長軸を a 、短軸を b とする楕円である (図 35)。この式をもとに2焦点が与えられたときの楕円の一般的な方程式を導出する。すると、求める楕円方程式は、式(13)を原点を中心に θ 回転し、 x 軸方向 p 、 y 軸方向 q 平行移動したものであると考えられる (図 36)。

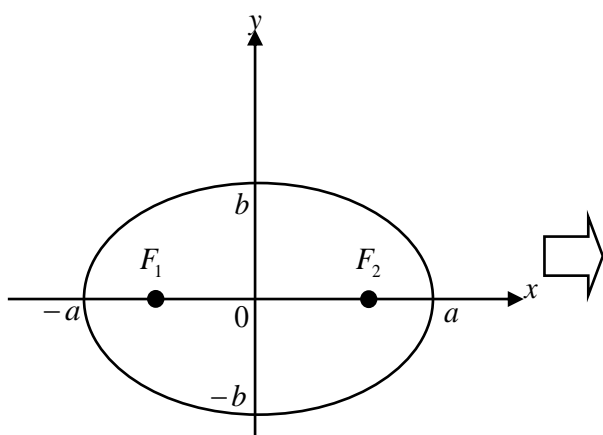


図 35. 高校で習う楕円
Fig.35 The ellipse formula
learned at a high school

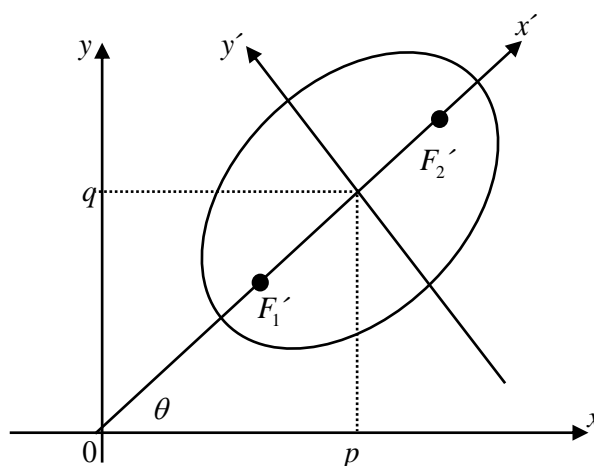


図 36. 高校では習わない楕円
Fig.36 The ellipse formula
which is not learned at a high school

そこで、式(14)に示す直交変換を行い、座標系を $x'-y'$ に移す。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x-p \\ y-q \end{pmatrix} \quad (14)$$

これにより、関数 $f(x, y)$ は

$$f(x', y') = f((x-p) \cos \theta + (y-q) \sin \theta, -(x-p) \sin \theta + (y-q) \cos \theta) \quad (15)$$

となる。

以上より求める楕円方程式は、式(16)で表せる。

$$\frac{((x-p)\cos\theta+(y-q)\sin\theta)^2}{a^2} + \frac{(-(x-p)\sin\theta+(y-q)\cos\theta)^2}{b^2} = 1 \quad (16)$$

ここで、 p , q , $\cos\theta$, $\sin\theta$, a , b の値は分からないため、与えられた焦点から算出する。その手順を以下に示す。

- 1) 2焦点 $F_1(x_1, y_1), F_2(x_2, y_2)$ を得る。
- 2) 原点から x 軸方向 p , y 軸方向 q の平行移動を求める。

$$p = \frac{x_1 + x_2}{2}, q = \frac{y_1 + y_2}{2} \quad (17)$$

- 3) 回転角度 θ を求める。

2直線のなす角より $\cos\theta$, $\sin\theta$ を求める。これらの値は、式(18)で示した2直線が与えられたときに式(19)によって求めることができる。

$$\begin{cases} a_1x + b_1y + c_1 = 0 \\ a_2x + b_2y + c_2 = 0 \end{cases} \quad (18)$$

$$\cos\theta = \frac{a_1a_2 + b_1b_2}{\sqrt{a_1^2 + b_1^2}\sqrt{a_2^2 + b_2^2}}, \sin\theta = \frac{a_1b_2 - a_2b_1}{\sqrt{a_1^2 + b_1^2}\sqrt{a_2^2 + b_2^2}} \quad (19)$$

求める角度は、 x 軸($y=0$)とのなす角を求めればよいので、係数は式(20)となる。ここで、 $x_1 = x_2$ のとき、 a_2 の分母は 0 になってしまう。しかし、このとき回転角度は明らかなので、これらの式は使わずに求めることができる。

$$a_1 = 0, \quad b_1 = 0, \quad a_2 = \frac{y_2 - y_1}{x_2 - x_1}, \quad b_2 = -1 \quad (20)$$

- 4) 楕円の長軸 a , 短軸 b を求める。

a はアバタの最高速度と更新間隔より適当な値を定める。 b の値は式(21)で求める。

$$b = \pm\sqrt{a^2 - c^2} \quad \text{※}c = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{2} \quad (21)$$

ここで、 c は焦点間の距離を 2 で割った値を使用している。

以上の手順で求めた各値を式(16)に代入して楕円の方程式を求める。