

2002 年度 卒 業 論 文

集団移動シミュレーションにおける
移動障害物の回避に関する研究

指導教員：渡辺 大地 講師
和田 篤 講師

メディア学部 3DCG アプリケーション構築プロジェクト

学籍番号 99p430

森江 修也

2003年3月

2002年度 卒業論文概要

論文題目

集団移動シミュレーションにおける
移動障害物の回避に関する研究

メディア学部
学籍番号: 99p430

氏名

森江 修也

主査

渡辺 大地 講師

副査

和田 篤 講師

キーワード

回避 シミュレーション boid モデル A* アルゴリズム

近年、コンピューターの性能向上により、ゲームなどにおいて、多数のオブジェクトを同時に扱うことが可能になった。しかし、仮想空間上で多数のオブジェクトを配置して扱うといった場合に、ユーザーが一つ一つ操作するのは煩雑である。オブジェクトが集団として自動的に接触や衝突を回避し、スムーズに目的地まで移動してくれる手法があれば、そういった操作を軽減することができる。

本研究では、汎用的な障害物回避を目指し、集団移動において、移動する障害物を回避し、目的地まで移動するモデルの実現を目標とするものである。

具体的な手法は次のとおりである。回避方法は従来の経路検索で多く用いられるメッシュ分割によるものではなく、集団を構成するキャラクターと、障害物との相対的な位置、及び進行方向から随時判断して回避する方法を採用した。相対的な位置関係は、距離と、キャラクターの進行方向と障害物のある方向との角度の2点から判断し、同時に障害物の進行方向も考慮する。集団制御にはboidモデルを採用した。boidモデルは簡単なルールで集団制御を行うことができると言う点で優れており、利用例が非常に多い。また、問題の条件を次のように限定した。集団を構成する個体は全て同じ大きさとした。移動障害物の形状は単純なものとし、大きさについては個体とおおよそ近い大きさとした。障害物の移動は等速直線運動とし、進路転換を行わないものとした。

相対的な位置関係からの移動障害物回避を実装し、機能確認を行った結果、集団同士での回避はほぼ確実に行われ、ランダムな移動障害物も多くの場合で回避できることが確認できた。

目次

第1章 研究の概要と動機	1
第2章 集団制御法	2
第3章 既存の回避手段と問題点	5
第4章 移動障害物回避	7
4.1 相対的位置関係からの静止障害物回避	7
4.1.1 基本回避	7
4.1.2 緊急回避	9
4.2 相対的位置関係からの移動障害物回避	10
4.2.1 基本回避	10
4.2.2 緊急回避	14
4.3 集団での回避	20
4.4 集団同士の回避	20
第5章 評価	21
第6章 まとめ	25
謝辞	26
参考文献	27

第1章 研究の概要と動機

近年、コンピューターの性能向上により、インタラクティブコンテンツなどにおいて、多数のオブジェクトを同時に扱うものが増えてきている。例えばゲームなどである。しかし、そのようなコンテンツでオブジェクトを扱う際に、ユーザーが全てのオブジェクトを一つ一つ操作するのは煩雑である。キャラクタが集団として自動的に接触や衝突を回避し、スムーズに目的地まで移動してくれる手法があれば、そういった操作を軽減することができる。これまでの研究では、決められた地形上で障害物と接触しないような目的地までの経路を見つけ出す、経路プランニングに関する研究はあった[9, 10, 11, 12]。A*などが代表的である[3, 4]。しかしながら、動いている障害物を回避しながら目的地まで向かうというようなものはまだあまり研究されてない。そこで、本研究では汎用的な障害物回避を目指し、集団移動において、移動する障害物を回避し、目的地まで移動するモデルの実現を目標とするものである。

具体的な手法は次のとおりである。回避方法は従来の経路検索で多く用いられるメッシュ分割によるものではなく、集団を構成するキャラクタと、障害物との相対的な位置、及び進行方向から随時判断して回避する方法を採用した。相対的な位置関係は、キャラクタと障害物との距離と、キャラクタの進行方向を基準とした障害物のある方向の2点から判断し、同時に障害物の進行方向も考慮する。集団制御にはboidモデルを採用した[1, 2]。boidモデルは簡単なルールで集団制御を行うことができると言う点で優れており、利用例が非常に多い[5, 6, 7, 8]。また、boidモデルは個体間の位置関係を基準に集団制御を行っており、本研究で提案する回避法との組み合わせに適している。

また、問題の条件を次のように限定した。集団を構成する個体は全て同じ大きさとした。移動障害物の形状は単純なものとし、大きさについては個体とおおよそ近い大きさとした。障害物の移動は等速直線運動とし、進路転換を行わないものとした。

最終的な評価では、ランダムに配置される移動障害物の集団での回避と、集団同士での衝突回避を行う。ランダムに配置される障害物では、さまざまな、予測しない状況で正しく回避が機能するかどうかの評価を行い、逆に、集団同士での回避は同じアルゴリズムで動くもの同士が正しく回避できるかどうか、また、スクランブル交差点のような複数の集団の流れが存在する場面で衝突することなくスムーズに抜けることができるかを評価する。

第2章 集団制御法

本研究では集団の制御にboidモデルを採用した。

Boidモデルは、3つのルールだけで集団をシミュレーションできるというモデルである。このboidという単語は、バード(鳥)とアンドロイドを組み合わせた造語だが、鳥に限らずさまざまな集団の表現法として用いられている。本研究でも、集団の制御が容易で、かなり集団らしくみえることから、集団制御法として採用した。

Boidモデルで使用される3つのルールは次のとおりである。

Separation(引き離し) : 周りの仲間と一定の距離を保つ

Alignment(整列) : 他の仲間と進行方向を合わせる

Cohesion(結合) : 集団の中心に向かう

このルールは一定領域内に入ったキャラクタに対して働く。ここでキャラクタとは、集団を構成する個々の個体を指す。この領域の範囲は、どういった集団を表現するかによって変わってくる。範囲を広くしていくと整然とした集団になり、集団内でのキャラクターの位置関係が変わったり、集団がばらついたりといったことが起こりにくくなる。逆に狭くすると、集団全体の形が変わったり位置が入れ替わったりといったことが起こりやすくなる。

各ルールについての説明を以下に述べる。

Separation(引き離し)

通常、われわれは歩くときなど、周りの人とちょうどよい距離を保とうとする。Separationは、領域内に入った他のキャラクタと衝突しないように距離を開けようとするルールである。領域内に入ったすべての仲間と、自分との相対的な位置ベクトルの和を求め、その逆ベクトルの方向へ移動する。図2.1がその様子で、緑色の三角形が自分、青い三角形が仲間、灰色の円が作用する領域。緑色の直線が領域内の仲間の方向で、この場合赤い矢印の

方に移動する。

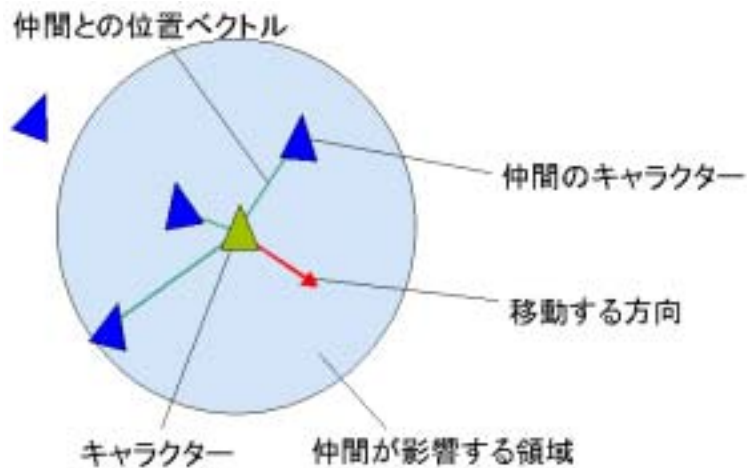


図2.1 Separation

Alignment(整列)

Alignmentは、領域内にいる他のキャラクタと、進行方向を合わせようとするルールである。領域内にいる他のキャラクタの、進行方向角の平均を取り、自分の進行方向をその方向に回転する。図2.2がその様子で、それぞれのキャラクターを表す三角形から伸びる直線が進行方向を示し、赤い矢印の方に進行方向を修正している様子である。

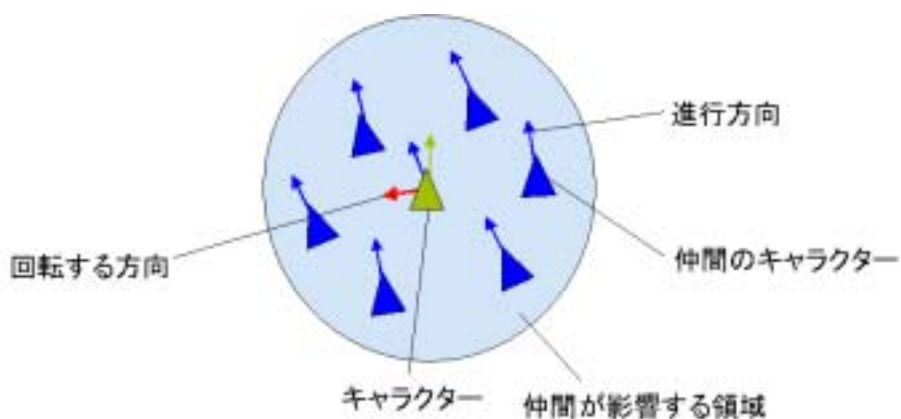


図2.2 Alignment

Cohesion(結合)

Cohesionは、集団から離れてしまわないために、領域内にいる他のキャラクタの中心に向

かおうとするルールである。領域内にある他のキャラクターの、座標の平均から中心を求め、中心に向けて移動する。図2.3は、緑色の点が領域内の仲間の中心で、赤い矢印のように移動する様子である。

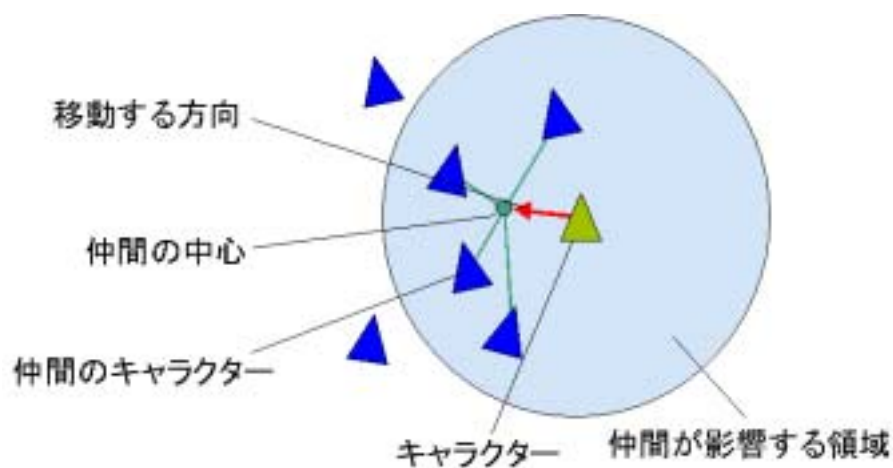


図2.3 Cohesion

実装する段階では、これら3つのルールのうち Separation と Cohesion により得られたベクトルの和をキャラクターの移動とし、進行方向をAlignment で得られた方向することで集団が形成される。

第3章 既存の回避手段と問題点

本章では、既存の経路プランニング法と、移動障害物回避へ応用する際に生ずる問題点を述べる。

A*をはじめとする既存の経路プランニング法では、対象空間のモデル化に際し、通常、正方形の均等なメッシュ状に環境を分割し、隣接するメッシュへの移動の連続により経路を表現する。

A*アルゴリズムでは、ゴールまでの最短の経路を一気に検索し、実際に動くときはあらかじめ導き出されたルートを通る。実際にA*を用いて経路プランニングを行った例を図3.1に示す。黒く塗りつぶされたメッシュが、障害物の存在する部分で、STARTからGOALまで繋がっている線が経路を示している。これは、決められた地形からゴールまでの最短な経路を求めるには有効であるが、常に動く障害物では、スタートする瞬間の障害物の位置から経路を求めても障害物の位置が変わってしまい、求めた経路上を障害物が通れば衝突してしまうので、明らかに無理がある。

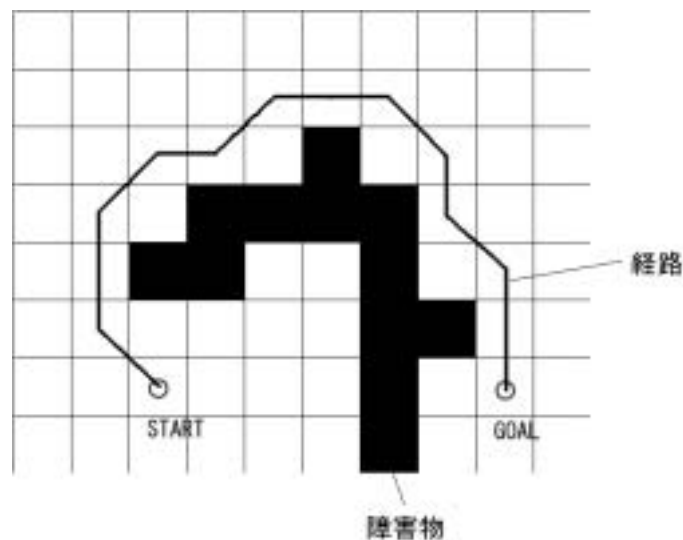


図3.1 A*による経路検索の例

Real-Time A*アルゴリズムは、経路の検索を行いつつ目的地への移動を行うものである。

A*との違いは、経路検索と移動とを交互に行うことで、A*よりも状況の変化に対応しやすい。これは、移動障害物の進行方向と同じ方向に回避しようとして衝突したり、進行方向が平行になればいつまでも同じ方向に進み続けてしまう問題点がある。

これらの事から、既存の回避手段で移動障害物が困難な要因としては、障害物の進行方向を考慮していない点にあると思われる。また、対象空間のモデル化に際してメッシュ分割をもちいて経路を検索しようとしているが、モデルの移動障害物の移動量や、進行方向によっては、メッシュが大きすぎるとは移動不可能な範囲が必要以上に広がってしまい、より細かく分割する必要が出てくるが、それでは計算量が増え、効率が悪くなると思われる。

第4章 移動障害物回避

本章では実際に移動障害物を回避する方法について提案する。まず、静止している障害物を相対的位置関係から回避する方法を述べ、次に移動障害物の回避が出来るように拡張した手法について述べる。

4.1 相対的位置関係からの静止障害物回避

4.1.1 基本回避

ここでは、基本的な障害物の回避法について述べる。まず、対象となる障害物との距離と、進行方向に対する角度を求める。そして、キャラクターの進行方向の一定範囲内に障害物が存在する場合、障害物の無い方へ進行方向を変更することで衝突を回避する。概念図を、図4.1に示す。

具体的な手順を以下に述べる。キャラクターの進行方向を \vec{v} 、キャラクターから見た障害物の相対位置を \vec{r}_o とし、 $d=|\vec{r}_o|$ 、 \vec{r}_o が \vec{v} に対して成す角を θ とする。一定範囲を、距離が k 、角度を進行方向の左右 $\pm\alpha$ 以内とし、 $d < k$ かつ $|\theta| < \alpha$ のとき、 $-\theta$ 方向に \vec{v} を回転させ、回避する。これを最も基本的な回避とし、基本回避と呼ぶことにする。位置関係を図4.2に、一定範囲を図4.3に示す。また、回避の例を図4.4に示す。

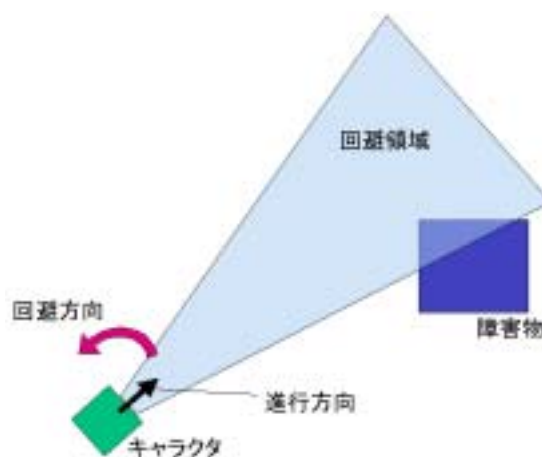


図4.1 基本回避の概念

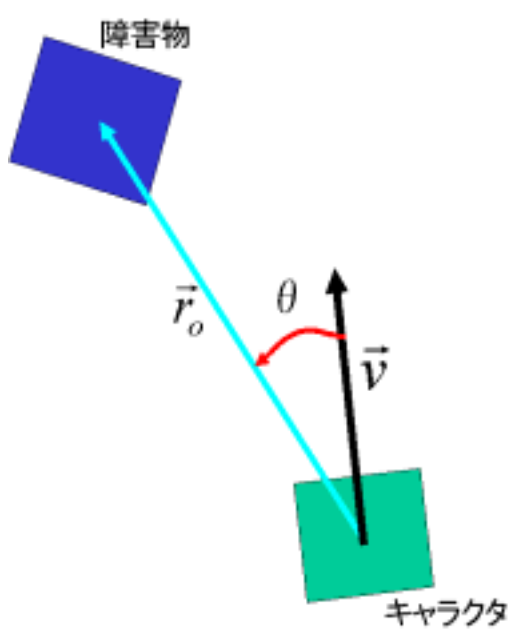


図 4.2 キャラクタと障害物の位置関係

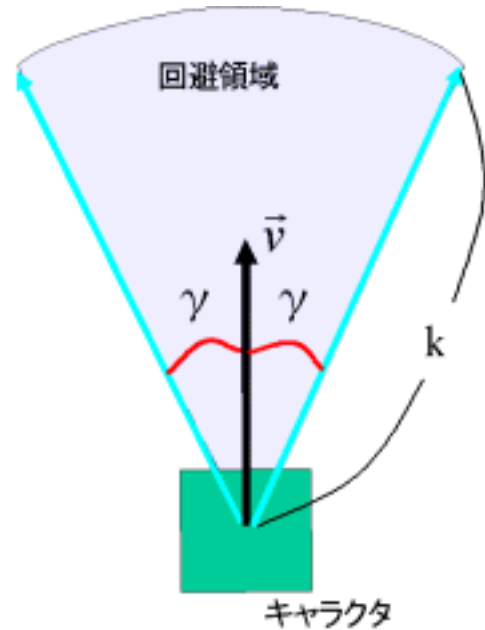


図 4.3 回避検出領域

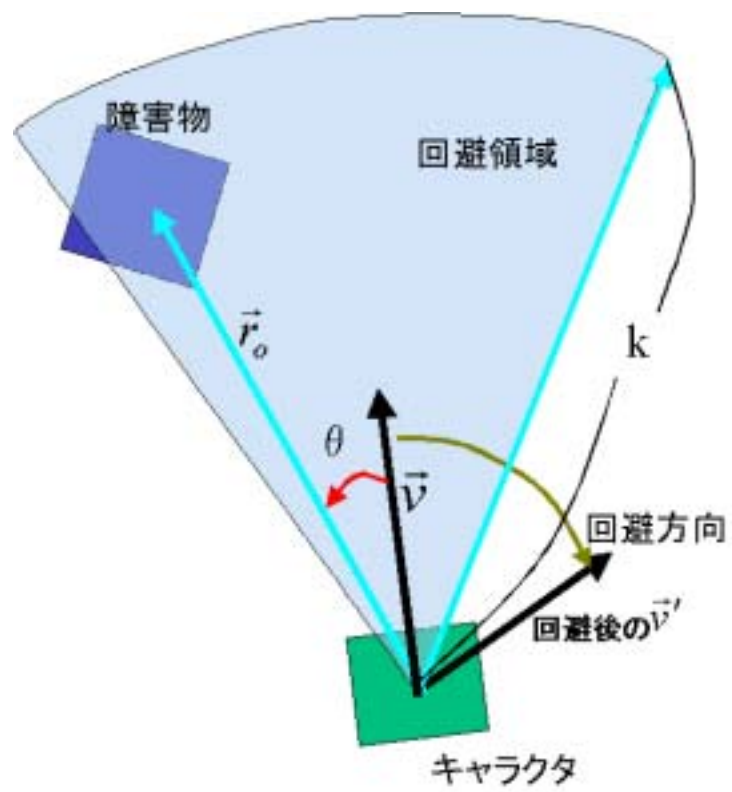


図 4.4 基本回避

4.1.2 緊急回避

基本回避だけでは前方にたくさんの障害物があったり、想定しない位置に障害物があった場合に対応できない可能性がある。例えば、いくら進行方向を変えても回避先に障害物がある場合などである。その問題を解決するため、緊急回避を加えた。前方一定距離内に障害物が存在する場合、強制的に左右に移動して障害物を回避する。これならば進行方向の転換よりもすばやく避ける事ができる。緊急回避の概念図を図4.5に示す。

具体的な手順を述べる。緊急回避を行う基準となる距離を m とし、 $d < m$ ならば、左右の障害物が無い方へ避ける。 \vec{v} に対して垂直なベクトルを \vec{v}_n とし、 $\vec{v}_n \cdot \vec{r}_o > 0$ ならば \vec{v}_n 、 $\vec{v}_n \cdot \vec{r}_o \leq 0$ ならば $-\vec{v}_n$ に移動して回避する。これを図4.6に示す。

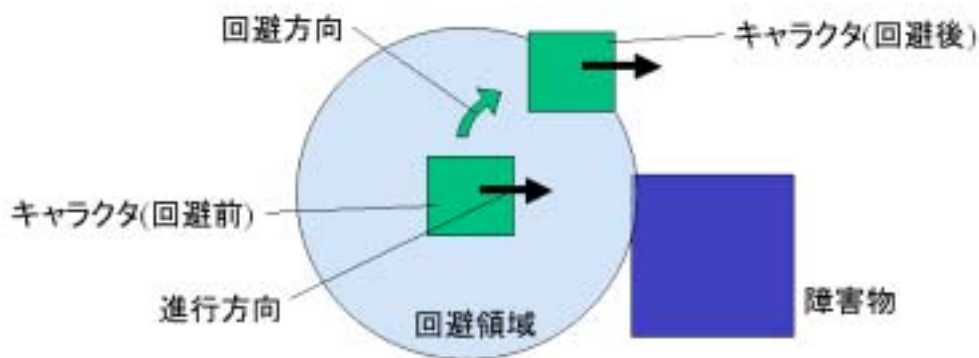


図4.5 緊急回避の概念

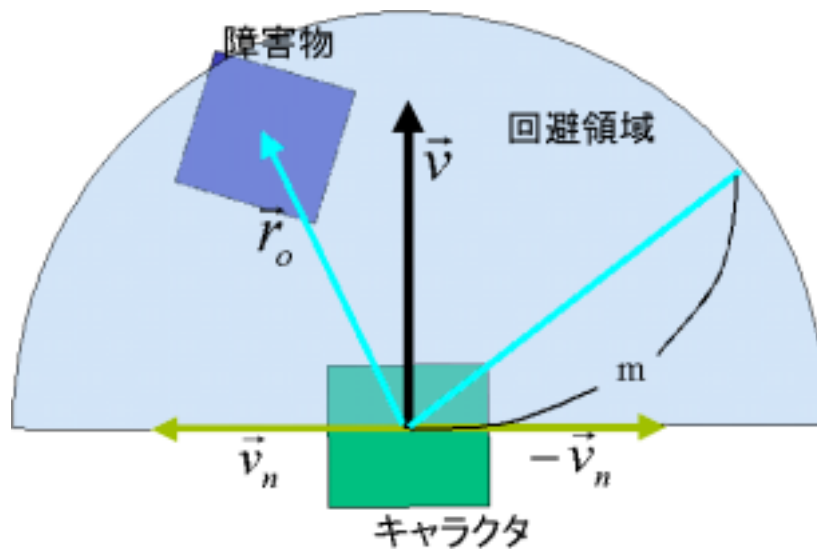


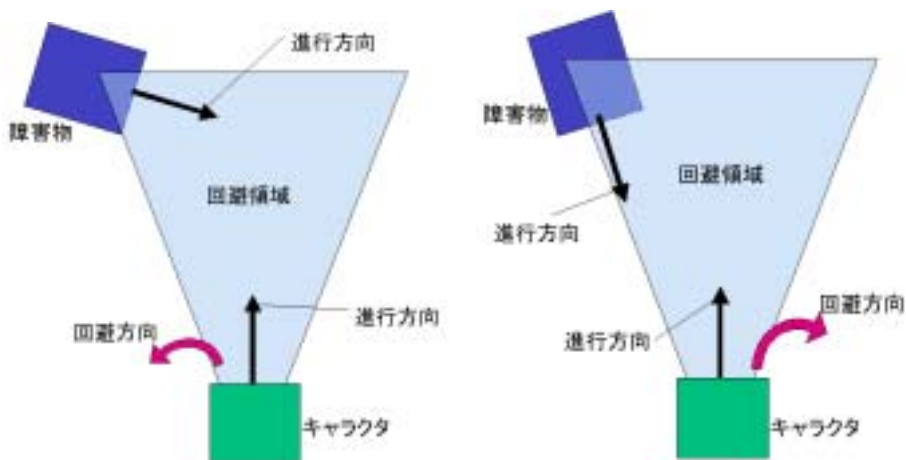
図 4.6 緊急回避

4.2 相対的位置関係からの移動障害物回避

4.2.1 基本回避

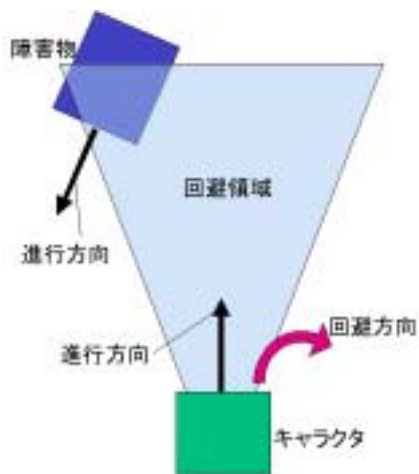
次に、移動する障害物の回避について述べる。静止障害物と移動障害物の違いは当然ながら移動していることである。それを効率よく回避するためには、位置関係だけでなく、障害物の相対的な進行方向を考慮することが有効である。この場合、相対的な移動方向から、障害物の移動方向とは逆の方向に進行方向を変える事で効率よく回避することができる。静止障害物と移動障害物の回避手段の特徴的な違いは、障害物の進行方向とキャラクターの進行方向が交差する場合である。また、相対的な進行方向が同じでも、キャラクターから見て障害物が右にあるか左にあるかで避け方は変える必要がある。そのことから、効率よく回避するためにパターン 1、2、3 の3つのパターンに分けた。パターン 1 は、一定領域内でキャラクターの進行方向に対して垂直に近い角度で接近してくる障害物を避けるパターンで、ここでは障害物のある方向に進路転換を行う。もしも逆に進路転換を行った場合、障害物と同じ方向に進むことになってしまい、衝突の可能性がある。パターン 2 は、進行方向に対して水平に近い角度で向かってくる障害物で、この場合、障害物の無い方に避ける。もしここでパターン 1 と同じ方向に進路転換した場合、正面から衝突して

しまう可能性がある。パターン は、距離は近づいているがキャラクターの進行方向とは交差しない方向に進む障害物のパターンである。これも、障害物の無い方に避ける。この場合は、静止障害物の回避と同様の基本回避でよい。各パターンの概念図を図 4.7 に示す。また、 ~ 以外の、キャラクターに接近して来ない場合は、自然に距離が離れ、衝突する可能性は低いので回避行動をとる必要は無い。



(a)パターン : 左から接近する障害物を回避するためにキャラクターを左に進路転換させている

(b)パターン : 左から接近する障害物を避けるために、障害物の無い右に避けている



(c)パターン : 障害物の無い右方向に進路転換して回避する。

図 4.7 移動物体に対する基本回避のパターン ~ の概念図

具体的な手順は次のようにする。障害物の進行方向を \vec{o} 、 \vec{v} と \vec{o} の成す角を α とする。また、基準となる角度を β とする。図 4.8 にキャラクタと障害物の相対的な進行方向、図 4.9 に基準となる角を示す。

$d < k$ かつ $\alpha < \beta$ の場合、キャラクタは回避を行う。回避パターンは α によって 3 通りに分かれる。 $\beta/2 < \alpha < \beta$ の場合は + 方向に回避、 $\alpha < \beta/2$ の場合は - 方向に回避、 $\alpha > \beta$ の場合は - 方向に回避する。各々の回避パターンを図 4.10、4.11、4.12 に示す。

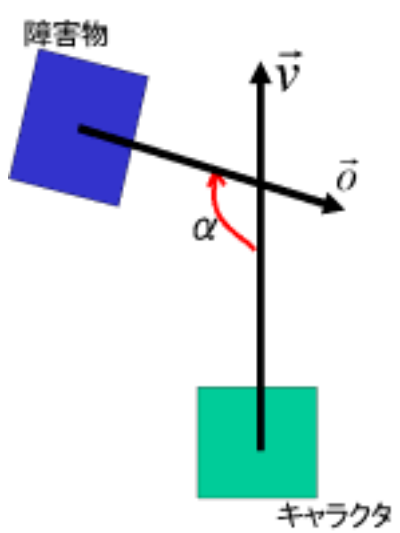


図 4.8 キャラクタと障害物の相対的な進行方向

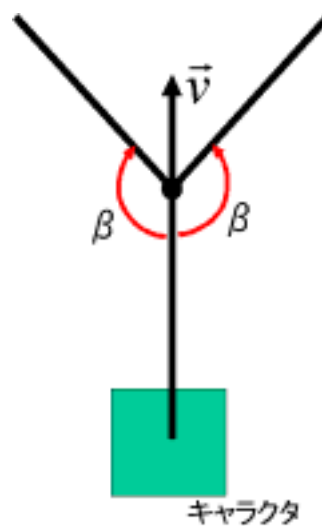


図 4.9 回避の基準となる角

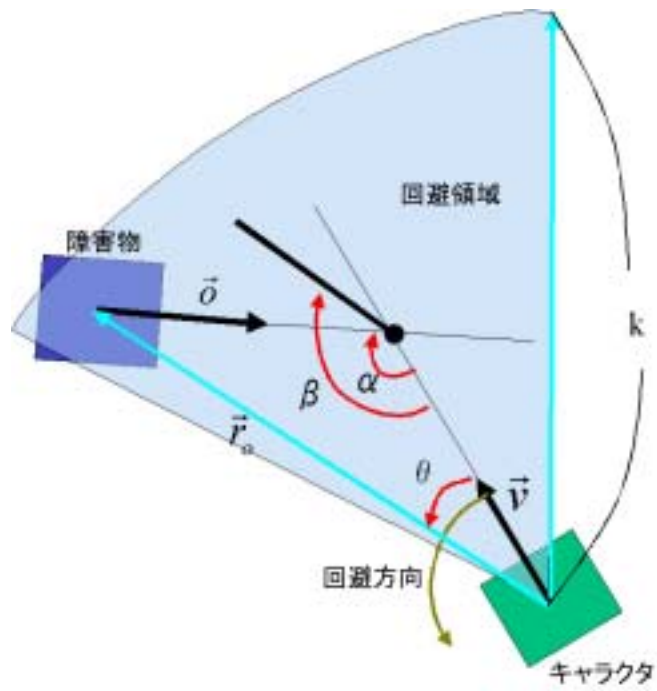


図 4.10 基本回避パターン

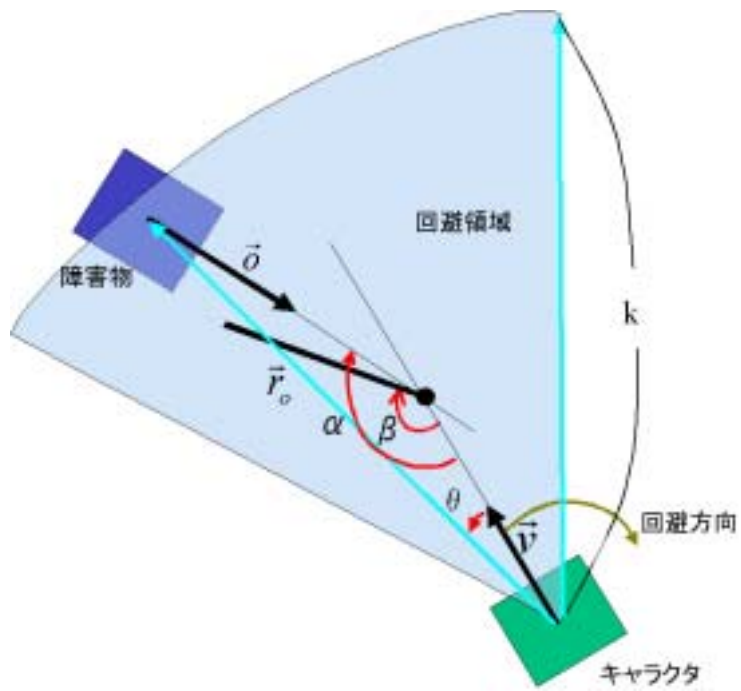


図 4.11 基本回避パターン

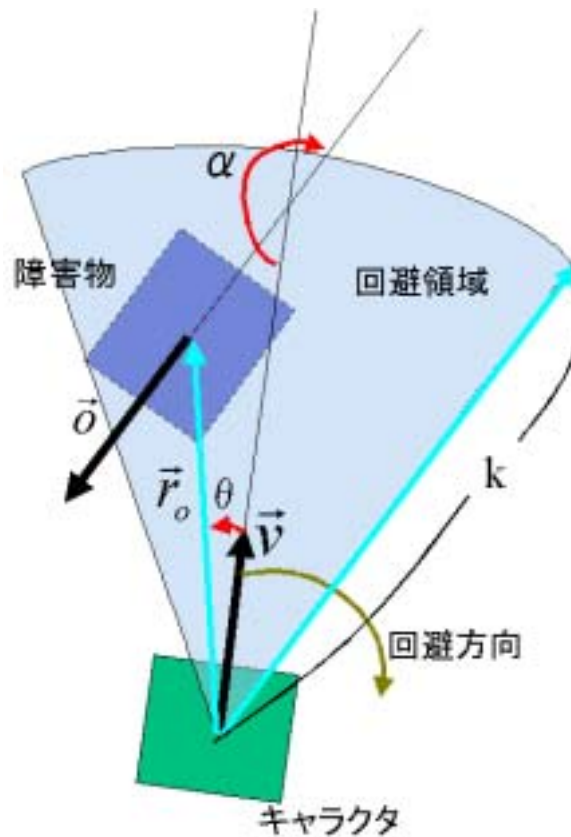
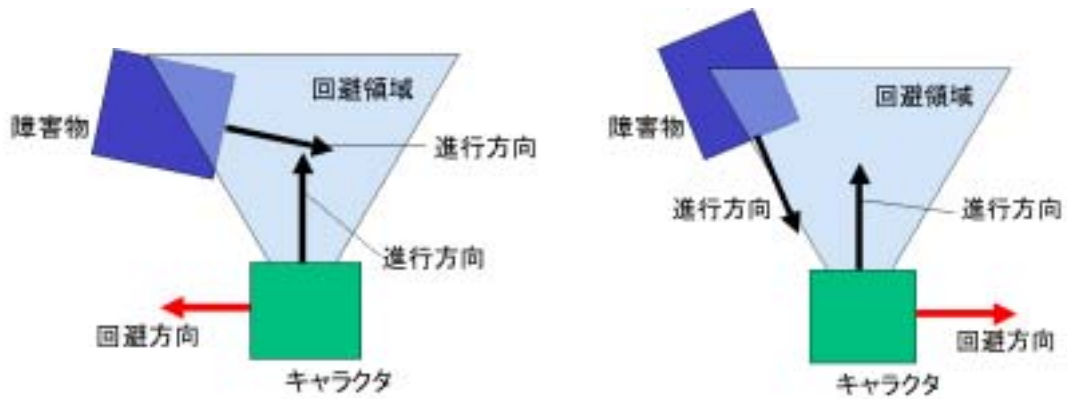


図 4.12 基本回避パターン

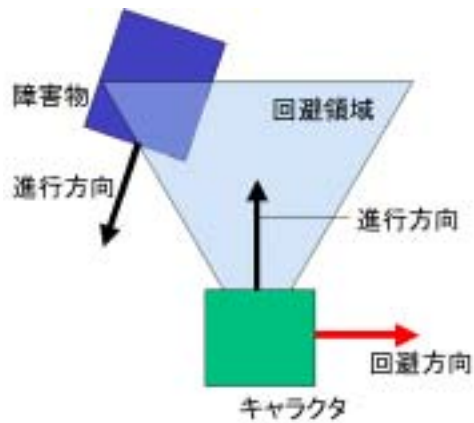
4.2.2 緊急回避

次に緊急回避について述べる。移動障害物でも静止障害物の緊急回避と同様に、基本回避では回避できなかった障害物をキャラクタを強制的に左右に移動することで回避する。緊急回避においても障害物の進行方向を考慮して回避する。効率的に回避するために4つのパターンに分ける。まず、3つの緊急回避のパターンを示す。このパターンを場合分けする基準は前述した基本回避のものと同様である。回避する方向は、 \vec{v} では、障害物がある方向に、 \vec{v}_0 と \vec{o} では、障害物が無い方向に回避する。概念を図4.13に示す。さらに4つめのパターンとしてキャラクターの左右から障害物が接近するパターンがある。静止障害物の場合は左右の障害物を考える必要は無かったが、移動障害物では衝突の可能性が発生するため、これも緊急回避の対象となる。この場合は、障害物が無い方へ回避する。概念を図4.14に示す。



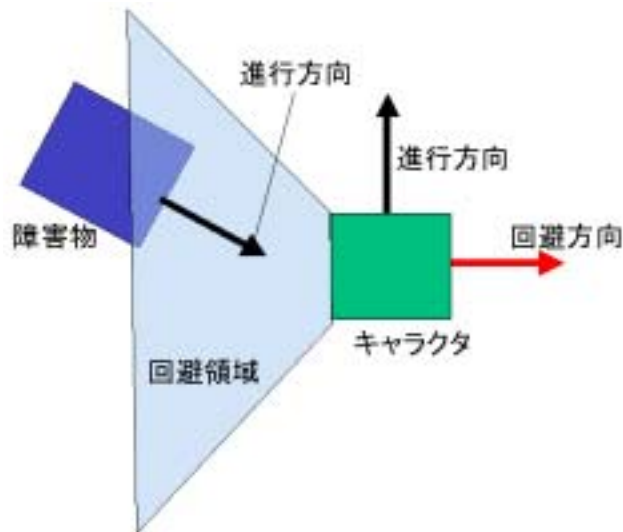
(a)パターン : 左から来る障害物を回避するために、障害物のある左方向に避けている

(b)パターン : 左から接近する障害物を回避するために、障害物の無い右方向に避けている



(c)パターン : 障害物の無い方へ避けている

図 4.1 3 移動障害物緊急回避のパターン ~ の概念



キャラクターの左横方向から接近してくる障害物を回避するため、右方向へ移動している。

図 4.1 4 移動障害物緊急回避パターン の概念

具体的な手順は次のようにする。緊急回避を行う範囲を、距離 m 、角度を進行方向の左右 θ_1 以内とする。 $d < m$ かつ、 $\theta < \theta_1$ のとき、パターン 1 の回避を行う。また、 $d < m$ かつ、 $\theta_1 < \theta < \theta_2$ のとき、パターン 2 の回避を行う。パターン 1 の回避領域を図 4.1 5 に、パターン 2 の回避領域を図 4.1 6 に示す。パターン 1 は、 $d < m$ かつ $\theta \leq \theta_1$ で、 $\theta < \theta_1$ の場合である。この場合は $\vec{v}_n \cdot \vec{r}_o > 0$ ならば $-\vec{v}_n$ 、 $\vec{v}_n \cdot \vec{r}_o \leq 0$ ならば \vec{v}_n に移動して回避する。これを図 4.1 7 に示す。パターン 2 は、 $d < m$ かつ $\theta \leq \theta_1$ で、 $\theta_1 < \theta < \theta_2$ の場合である。この場合は $\vec{v}_n \cdot \vec{r}_o > 0$ ならば \vec{v}_n 、 $\vec{v}_n \cdot \vec{r}_o \leq 0$ ならば $-\vec{v}_n$ に移動して回避する。これを図 4.1 8 に示す。パターン 3 は、 $d < m$ かつ $\theta \leq \theta_1$ で、 $\theta \geq \theta_2$ のときである。この場合は $\vec{v}_n \cdot \vec{r}_o > 0$ ならば \vec{v}_n 、 $\vec{v}_n \cdot \vec{r}_o \leq 0$ ならば $-\vec{v}_n$ に移動して回避する。これを図 4.1 9 に示す。パターン 4 は、 $d < m$ かつ $\theta_1 < \theta < \theta_2$ で、 $\theta < \theta_2$ のときである。この場合は $\vec{v}_n \cdot \vec{r}_o > 0$ ならば \vec{v}_n 、 $\vec{v}_n \cdot \vec{r}_o \leq 0$ ならば $-\vec{v}_n$ に移動して回避する。これを図 4.2 0 に示す。

ここまで述べた基本回避と緊急回避の各パターンを用いる事によって、移動障害物の回避が可能である。

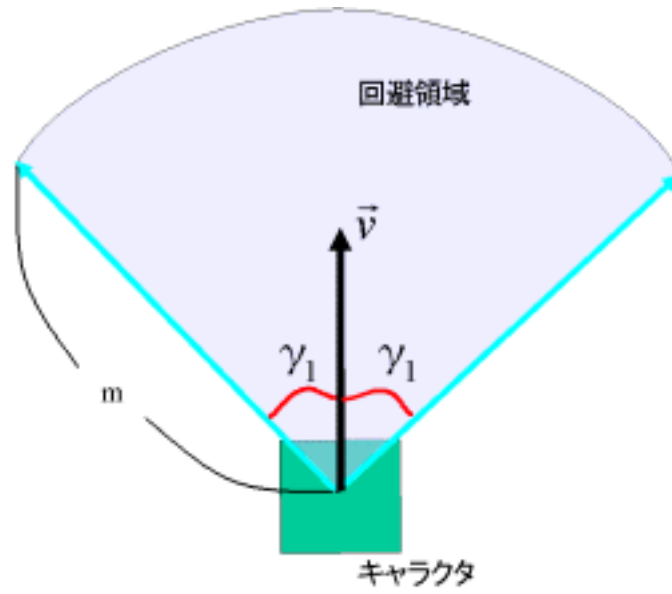


図 4.1 5 緊急回避パターン ~ の回避領域

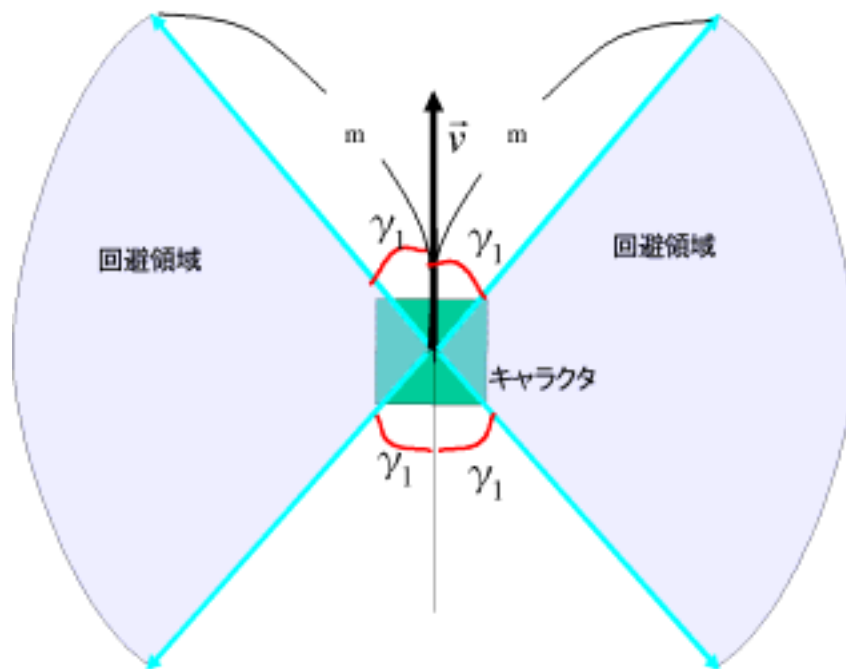


図 4.1 6 緊急回避パターン の回避領域

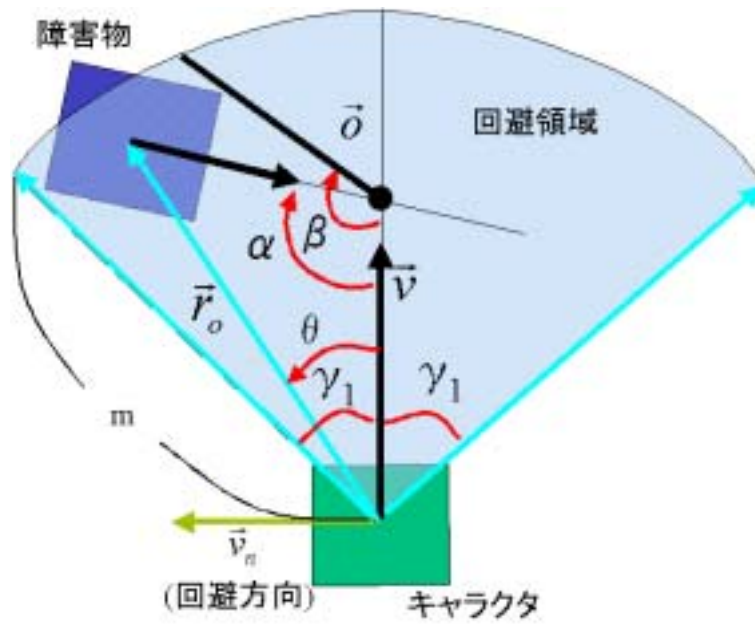


図 4.17 緊急回避パターン

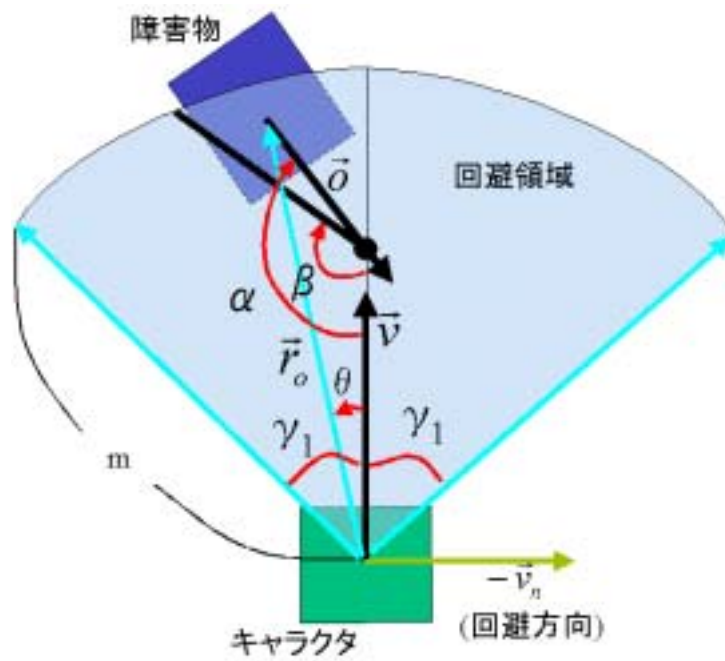


図 4.18 緊急回避のパターン

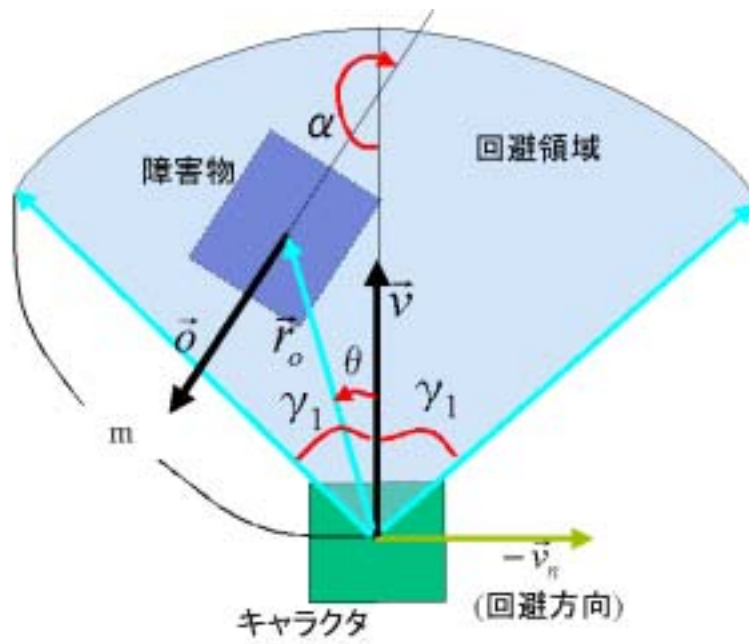


図 4.19 緊急回避のパターン

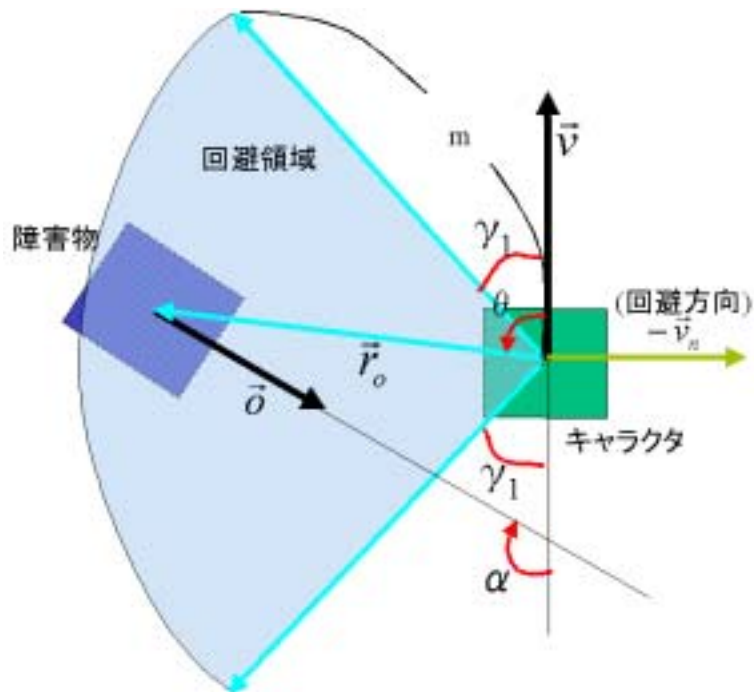


図 4.20 緊急回避のパターン

4.3 集団での回避

これまでに述べた移動障害物の回避法と、2章で紹介したboidを組み合わせる事で集団での回避を実現する。しかし、組み合わせるに際して問題が発生する。

集団維持の為に移動制御を行った際、その移動先に障害物が存在する場合、これと衝突してしまう。この問題点を解消するため、集団維持よりも障害物回避を優先的におこなうこととした。障害物回避の必要がある場合に集団維持を行わず、回避に専念する事で、確実に避けることができる。この場合、状況によっては集団が維持できずばらばらになってしまう可能性があるが、目的地へ向かう過程で集団は再構成される。

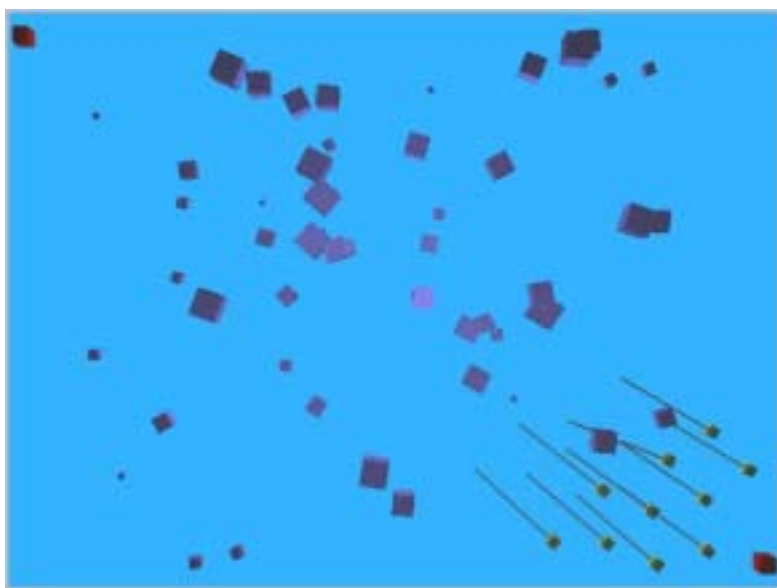
4.4 集団同士の回避

集団同士の回避はここまで述べた手法で可能である。障害物を、同じ集団の仲間ではない、他の集団のキャラクターと置き換えることで回避することが可能である。また、これまでの障害物と違い、集団同士であればお互いが衝突を回避しようとするため、よりスムーズに回避することが期待できる。

第5章 評価

本章では、これまでに述べた回避手法を実装し、実際に回避が正しく行われるか評価する。

図6.1のように、スタートとゴールを設定し、ランダムな方向に移動する障害物をランダムに配置し、集団がゴールまで障害物と接触することなく移動することが出来るかどうかを評価する。また、障害物の大きさも10段階でランダムに設定される。青いキューブが障害物で、緑色のキューブが集団を構成するキャラクター、また、キャラクターから伸びるラインは進行方向を示している。



青いキューブ：障害物

緑色のキューブ：集団を構成するキャラクター

右下の赤いキューブ：スタート

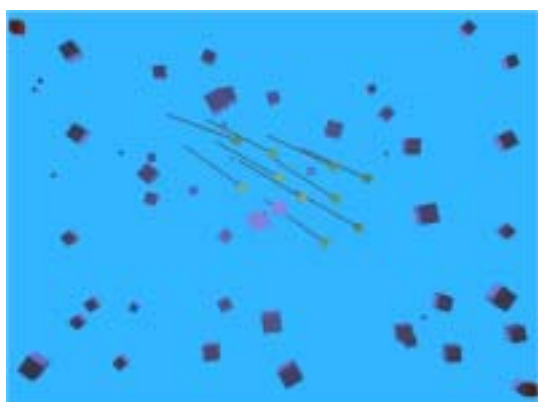
左上の赤いキューブ：ゴール

キャラクターから伸びるライン：進行方向

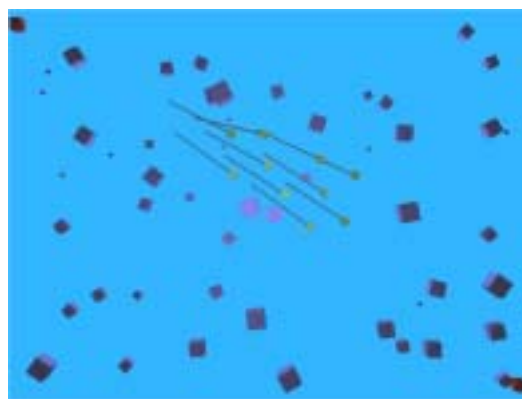
図6.1 スクリーンショット

図 6.2 が実際に動作している様子である。1 ~ 6 は、時間の流れを示している。図 6.2 に示した通り、多くの場合でこの回避アルゴリズムは有効に働き、障害物を回避した。しかし、障害物と接触してしまうパターンもあった。進行方向に極端に障害物が集中した場合や、後方からの接近による場合である。原因としては、この実装モデルでは、障害物は一定の方向にただ進むだけであることから、障害物同士の接触や、集団との接触などを一切考慮しないため、複数の障害物が集まって大きな障害物のような状態になったり、複数の障害物にはさまれるといった回避しづらい状況が発生するものと考えられる。

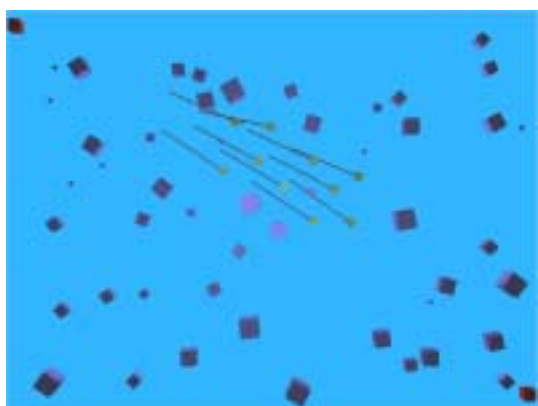
次に、集団同士の衝突回避を評価する。集団を構成する個体は全て同じ大きさ、同じスピードで、2つの集団が正面から交わるとき、衝突することなく相手集団を通り抜けることができるかどうか評価する。図 6.3 が集団同士での回避の様子である。集団 が青、集団 が緑のキューブで表されている。集団同士の回避では、確実に回避され、衝突は起こらなかった。同じアルゴリズムを使用するもの同士での回避であれば、お互いが相手との接触を回避しようとするため、より確実に回避できたものとする。



1



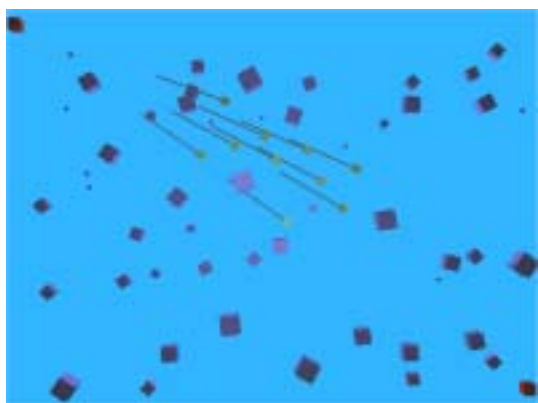
2



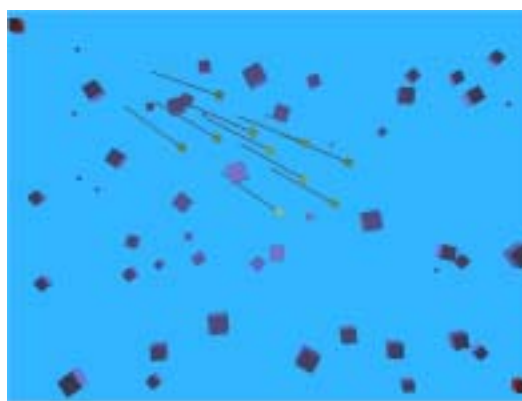
3



4



5



6

青いキューブ：障害物

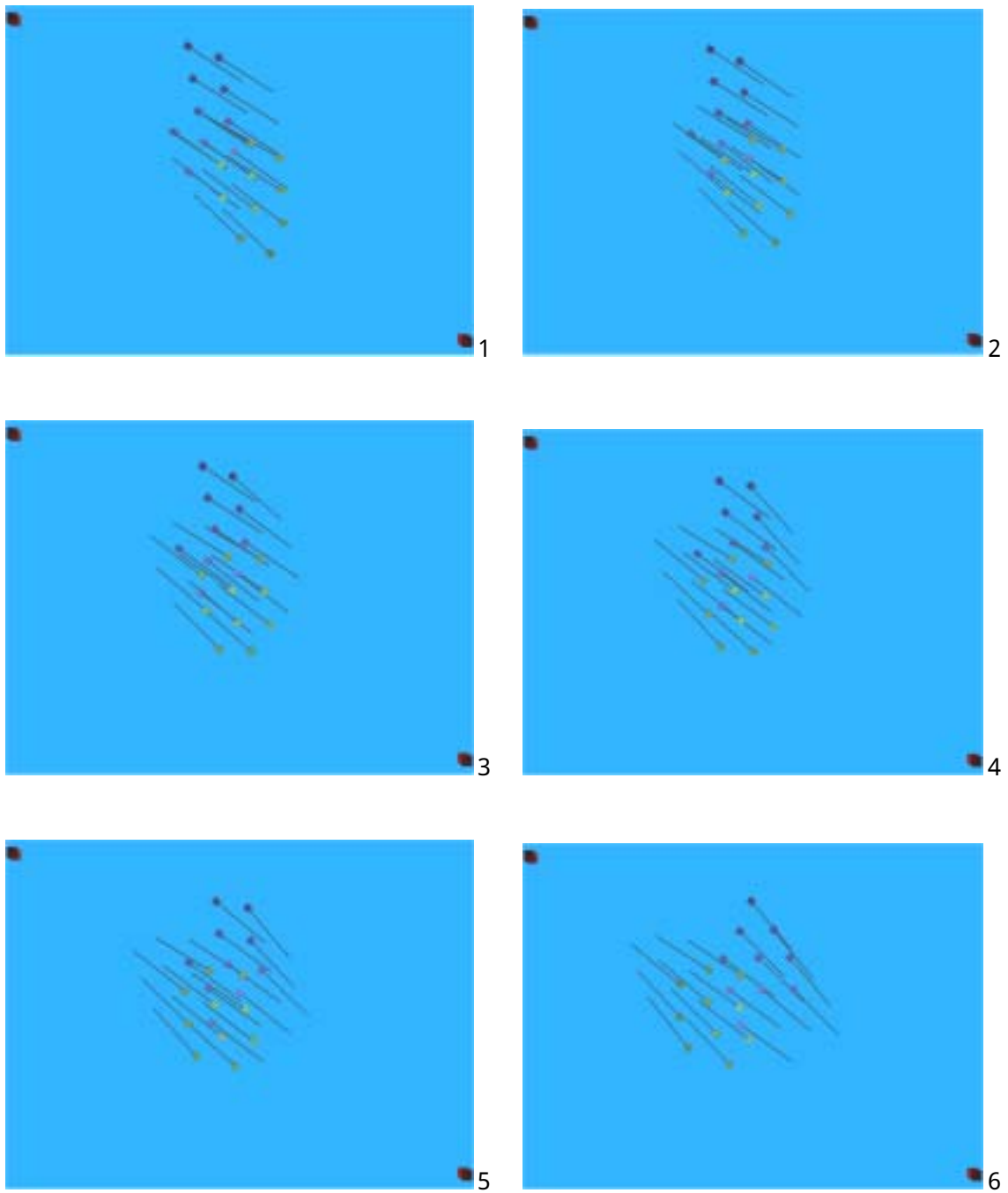
緑色のキューブ：集団を構成するキャラクター

右下の赤いキューブ：スタート

左上の赤いキューブ：ゴール

キャラクターから伸びるライン：進行方向

図 6.2 集団回避



青色のキューブ：集団

緑色のキューブ：集団

右下の赤いキューブ：集団のゴール

左上の赤いキューブ：集団のゴール

キャラクターから伸びるライン：進行方向

図 6.3 集団同士の回避

第6章 まとめ

近年、コンピュータの進歩にともなって、インタラクティブコンテンツは大きく発展し、さまざまなものが作られている。多数のオブジェクトを同時に扱うばあいには、集団での移動障害物回避手段は有用であると考えられる。そこでこの研究では、相対的な位置関係と、相対的な進行方向から移動する障害物を回避する手法を考えた。常に変化する状況で、衝突しないように目的地まで向かうためには多少集団が崩れても、回避を優先させて確実に回避を行うようにした。

相対的な位置関係からの移動障害物回避を実装し、機能確認を行った結果、集団同士での回避はほぼ確実に行われ、ランダムな移動障害物も多くの場合で回避できることが確認できた。また、リアルタイム性についても、現状ある程度高速な環境であれば利用可能であると思われる。

今後の課題としては、障害物の進行速度を考慮した回避や、より大きな移動障害物への対応などが考えられる。また、本研究の応用としては、A*等の経路プランニング法と組み合わせて、地形上の経路をA*で検索し、ポイント間の移動に本研究を用いて他のオブジェクトを回避させる、といったものが考えられる。

謝辞

本研究を進めるにあたり、終始温かいご指導をいただいた渡辺 大地 講師、和田 篤 講師に心よりの感謝の意を表します。

また、研究生生活において多々、お世話になりました事務の方々、メディア学部の諸先生方と学友に深く感謝いたします。

最後に、これまで温かく見守ってくださった家族に深く感謝します。

参考文献

- [1] Craig Reynolds, “ Flocks, Herds, and Schools : A Distributed Behavioral Model ”、SIGGRAPH ‘ 87
- [2] Craig Reynolds, “ Boids”、 <http://www.red3d.com/cwr/boids/>
- [3] Hart, P., Nilsson, N., and Raphael, B., “ A formal basis for the heuristic determination of minimum cost paths, ” IEEE Trans.Syst.Sci.,Cybern,Vol. ssc-4, No. 2,JULY, pp. 100–107, (1968).
- [4] Korf, R., “ Real-time heuristic search, ” Artif.Intell., Vol. 42, pp. 189–211,(1990).
- [5] Reza Olfati-Saber William B. Dunbar Richard M. Murray, “ Cooperative Control of Multi-Vehicle Systems using Cost Graphs and Optimization ” 、ACC03–IEEE0705
- [6] Christoph Niederberger, Markus H. Gross, “ Towards a Game Agent ”、 (2002)
- [7] Greg Schmidt, Michael Ringham, Donald House 、 “ Implementation of a Feature-based Gesture Recognizer with Application to a Physically-based Bird Animation System ” (1999)
- [8] 今給黎 隆, “ 水族館を作ろう:BOIDの基本 ”
http://if.dynsite.net/t-pot/program/101_boid1/
- [9] Brooks, R., “ Solving find path problem by a good representation of free space, ”、AAAI-82, pp. 381–386, (1982).
- [10] Lozano-Perez, T. and Wesley, M., “An algorithm for planning collision-free paths among polyhedral obstacles,” Comm. ACM, Vol. 22, pp. 560–570,(1979).
- [11] 田島稔幸, 中野幹, 市川雅也, 前田宏, “ 遺伝的アルゴリズムを用いたリアルタイム経路プランニング, ” 人工知能学会誌, Vol. 10, No. 1, pp. 94 104,(1995).
- [12] 登尾啓史, “ 未知空間におけるパスプランニングアルゴリズムを設計するための十分条件とそれにもとづくアルゴリズムの評価, ” 情報処理学会論文誌, Vol. 33, No. 8, pp. 1013 1021, (1992).