

コンフィギュレーション空間の低次元化によるキャラクター動作生成の効率化

野見山, 英登
九州大学システム情報科学研究院知能システム学部門

有田, 大作
九州大学システム情報科学研究院知能システム学部門

谷口, 倫一郎
九州大学システム情報科学研究院知能システム学部門

<http://hdl.handle.net/2324/5939>

出版情報 : 火の国情報シンポジウム, 2007-03
バージョン :
権利関係 :



コンフィギュレーション空間の低次元化による キャラクタ動作生成の効率化

野見山 英登† 有田 大作‡* 谷口 倫一郎*

九州大学大学院システム情報科学府†

九州大学大学院システム情報科学研究所*

〒819-0395 福岡県福岡市西区元岡 744

財団法人九州システム情報技術研究所‡

〒814-0001 福岡県福岡市早良区百道浜 2 丁目 1-22

E-mail: {nomiyama,rin}@limu.is.kyushu-u.ac.jp, arita@isit.or.jp

あらまし 本研究では、確率的ロードマップ法による三次元キャラクタの衝突回避動作生成について、モーションキャプチャデータシステムを用いて計測した人間動作の統計的な情報に基づいた効率化手法を提案する。本手法では、キャラクタの姿勢を表現するコンフィギュレーション空間に対して、モーションデータの主成分を用いた低次元化を行う。低次元化により、モーションデータから得られる人間の自然な姿勢のみを表現するコンフィギュレーション空間を構成し、この空間を用いることで、動作生成に用いるロードマップの作成を効率的に行う。従来の確率的ロードマップ法による衝突回避動作生成との比較実験を行うことで、提案手法の有効性を示す。

キーワード 3次元人物アニメーション, ロードマップ, 衝突回避, 経路設計

Effective Motion Planning by Dimensional Reduction of the Configuration Space

Hideto Nomiyama†, Daisaku Arita‡, Rin-ichiro Taniguchi*

Department of Intelligent Systems, Kyushu University †*

744, Motooka, Nishi-ku, Fukuoka, 819-0395, Japan

Institute of Systems & Information Technologies/KYUSHU‡

2-1-22, Momochihama, Sawara-ku, Fukuoka, 814-0001, Japan

E-mail: {nomiyama,rin}@limu.is.kyushu-u.ac.jp, arita@isit.or.jp

Abstract We propose an effective collision-free motion planning method based on probabilistic roadmaps for 3D character animation. The roadmaps are created by statistical information of human motions, which are observed by “a motion capture system”. We reduce the dimension of the configuration space by analyzing principal components of the motion capture data. By using this configuration space, we can efficiently create the roadmaps for motion planning. We confirm the effectiveness of our method by comparing with motion planning using the traditional probabilistic roadmaps method.

Key words 3D character animation, roadmaps, collision avoidance, path planning

1. はじめに

コンピュータグラフィックス (CG) 技術の発達により、シミュレーションシステムやコミュニケーションシステムにおいて、人間を模した CG キャラクタが用いられる場面が増えてきた。これに伴い、キャラクタに様々な動作を行わせるための研究が盛んに行われている。特に、衝突回避動作生成は様々な場面でキャラクタに自然な動きを行わせるために必要不可欠な技術である。これは、キャラクタに動作を行わせるとき、キャラクタ自身や周囲の障害物との間に衝突が発生しないようにその動作を生成する技術である。この衝突回避動作生成をキャラクタの体格や周囲の環境条件に合わせて手作業で行うことは非常に大きな労力が必要であり、衝突回避動作を自動的に生成できる技術が求められている。

衝突回避動作生成の研究に関しては、ロボット分野において既に多く研究がなされている。この分野では、ロボットの位置や各関節角度を一意に決定するパラメタ空間であるコンフィギュレーション空間 (C 空間) を用いる。この空間において、開始姿勢を表す点 q_{start} から終了姿勢を表す点 q_{goal} までの経路を設計することで、動作の生成を行う。特に、関節数の増加等によって C 空間が高次元空間となる場合の経路設計では、 C 空間内に標本点を取り、近接する標本点間をロボットが移動可能であれば枝で繋いでゆくことで経路を設計する“ロードマップ”を用いた確率的手法が有効とされている。

この確率的手法は、一般に single-query と multiple-query に大別できる。single-query では、一つの動作を生成するごとに逐次的に木構造のロードマップを作成してゆくことで、動的な環境に対応して動作経路設計を行うことができる。一方、multiple-query では、確率的ロードマップ法^[1]により C 空間全体にグラフ構造のロードマップを事前に作成しておくことで、動作生成時にはこのロードマップをたどるだけで様々な経路の設計を高速に行うことができる。これらの手法は CG 分野にも応用して用いられており、ロードマップを用いたキャラクタの衝突回避動作生成の研究も行われている^{[2][3][4][5]}。

実時間のシミュレーションシステムやコミュニ

ケーションシステムでは、円滑にキャラクタを動作させる必要があるためにキャラクタの動作は事前に作成されたものを用いることが多い。そのため、キャラクタが移動したり環境が変化した場合、障害物を無視した動作を行い、利用者に違和感を与えてしまうこととなる。この不自然さを取り除くことは重要であり、環境に応じた衝突回避動作を実時間で作成する必要性は高い。

この実時間という観点から、本研究では高速に動作の生成が行える確率的ロードマップ法を用いた実時間衝突回避動作生成を目指す。ただし、確率的ロードマップ法では、様々な動作を生成するために、 C 空間全体に多数の動作経路を作成しておく必要がある。そのため、 C 空間の次元が高くなるにつれ、指数関数的にロードマップ作成の計算量が増加してしまという問題がある。さらに、環境が変化した際にはロードマップの更新が必要となるという問題もある。

そこで本稿では、これらの問題を解決する手法として、モーションキャプチャデータを用いた C 空間の低次元化を提案する。低次元化により、モーションデータから得られる自然な姿勢のみを表現する C' 空間においてロードマップを作成することとなり、計算量の削減を行うことが可能となる。また、従来の確率的ロードマップ法による衝突回避動作生成との比較実験を行うことにより提案手法の有効性を示す。さらに、提案手法によって作成したロードマップを用いることで、Kallmann らが提案するロードマップの更新法^[6]による動作生成が頑健に行えることを実験によって示し、動的な環境下における高速な衝突回避動作生成の実現を行う。

2. 確率的ロードマップ法

2.1 コンフィギュレーション空間

コンフィギュレーション空間 (C 空間) とは、キャラクタの位置や各関節角度を一意に決定するパラメタ空間であり、 C 空間内の 1 点 1 点がそれぞれ固有の姿勢を表現する (図 1)。つまり、 C 空間内の点 q は、キャラクタの三次元位置 T と各関節角度 R によって表現される多次元点 $(T, R^{head}, R^{righthand}, R^{lefthand}, \dots)$ となる。 C 空間の次元数はキャラクタの自由度の数であり、キャラクタの可動する関節数が増加するごとに

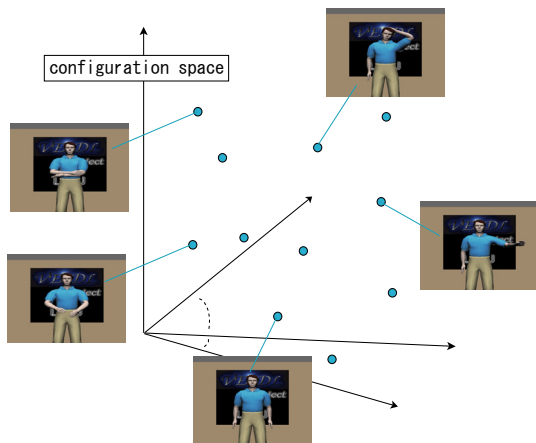


図 1: コンフィギュレーション空間

C 空間の次元数も増加してゆくこととなる。

この C 空間に対して、衝突を起こしておらず、かつ各関節角が制限角度内であるキャラクタの姿勢を射影した点が存在する部分空間を C_{free} とする。このとき、この部分空間 C_{free} 内のみを通る経路を設計することができれば、その経路が衝突回避動作となる。

2.2 ロードマップ

C_{free} 空間において経路を作成する際、この経路を連続する点で表現すると、 C 空間の広さからその情報は膨大なものになってしまう。このため、いくつかの標本点と近接する標本点間を結ぶ線分で構成されるデータ構造である“ロードマップ”を用いることで、経路の設計を行っていく。このとき、この標本点と線分をそれぞれロードマップの節点と枝と呼ぶ。

確率的ロードマップ法では、動作生成の事前処理としてグラフ構造のロードマップ (N, E) を作成する。ここで、 N はロードマップの節点情報であり、 C_{free} 空間内においてランダムに選ばれた標本点群 $\{q_1, q_2, q_3, \dots\}$ を保持する。また、 E はロードマップの枝情報であり、2 節点 $q_i, q_j (i \neq j)$ 間の接続情報を保持する。2 節点間の接続は、この 2 節点間を結ぶ線分が C_{free} 空間内に存在し、かつその長さがある閾値 TH_{dist} 以下である場合に行われる。

ロードマップの作成法については、Geraerts らが手法の比較を行っており^[6]、このような確率的

手法の有効性が示されている。

2.3 動作生成

動作生成には、入力として動作の開始姿勢と終了姿勢を C 空間に射影した点 q_{start}, q_{goal} を用いる。これらの 2 点を結ぶ経路をロードマップを用いて設計することで、衝突回避動作を生成することができる。ただし、2 点 q_{start}, q_{goal} がロードマップの節点であるとは限らないため、2 点間の経路を設計する場合、2 点とロードマップを接続する処理と、ロードマップ上での経路探索処理の 2 つの処理を行うこととなる。

2 点とロードマップを接続する処理では、まず、 q_{start}, q_{goal} とそれぞれの点に最も近いロードマップの節点との接続を考える。ここで、これらのロードマップの節点を $N(q_{start}), N(q_{goal})$ とする。このとき、 $q_{start} - N(q_{start})$ 間と $q_{goal} - N(q_{goal})$ 間に衝突が発生していなければ、この 2 点間を接続する。もし衝突が発生していれば、次に近いロードマップの節点を $N(q_{start})$ 、または $N(q_{goal})$ とし、再び接続できるかどうかを判定する。この処理を繰り返すことで、 q_{start}, q_{goal} とロードマップとの接続を行う。ただし、この処理において、 q_{start}, q_{goal} と接続できるロードマップの節点 $N(q_{start}), N(q_{goal})$ が見つからないとき、この q_{start}, q_{goal} 間を結ぶ経路は存在しないこととなり、動作の生成を行うことはできない。

次に、ロードマップ上での経路探索処理では、前処理で得られた $N(q_{start}) - N(q_{goal})$ 間のグラフ経路探索を行い、経路 P を設計する。この結果、これらの処理によって得られた経路 $q_{start} - P - q_{goal}$ が衝突回避動作として出力される。

2.4 確率的ロードマップ法の特徴

C 空間を用いた経路設計では、ある標本点 q が C_{free} 空間にあるかどうかを判定する衝突判定処理に多くの計算時間を必要とする。このため、 q_{start} から q_{goal} へ向けて、衝突判定処理を行いながら経路を伸ばしてゆく手法では、経路設計に時間がかかるという欠点があった。

これに対し、確率的ロードマップ法では、事前処理としてのロードマップ作成処理と、本処理としての経路設計処理の 2 つの処理を用いて動作生成を行う。事前処理の中で衝突判定処理を行い、衝突のないロードマップを作成すること

によって、経路設計処理ではほぼロードマップのグラフ経路探索処理さえ行えばよいこととなる。そのため、様々な開始姿勢と終了姿勢の組に対して、高速に衝突回避動作を生成することができる。

しかし、経路設計処理では q_{start}, q_{goal} とロードマップを接続する必要がある。この接続処理では、 q_{start}, q_{goal} とロードマップの節点間において衝突判定処理が必要である。さらに場合によっては q_{start}, q_{goal} とロードマップの接続ができず、経路設計が不可能となる可能性がある。このような問題を起こさず、高速に様々な動作を生成するためには、理想的にはすべての C_{free} 内の点と衝突を起こさずに接続が可能であるロードマップを作成する必要がある。つまり、事前処理においてロードマップを C_{free} 空間内全体に十分に密に作成する必要がある。

また、動的な環境下では環境の変化に応じて C_{free} 空間が変化してしまうため、この変化に合わせて事前処理であるロードマップ作成をやり直す必要がある。このため、環境が変化するたびに多くの計算時間が必要であるという欠点をもつ。本研究では、動的環境に対応するため、Kallmannらが提案するロードマップ更新法^[5]を基とした衝突回避動作生成システムを提案する。Kallmannらの手法では、事前にキャラクタの衝突情報を取得しておくことで、高速なロードマップ更新を実現している。

3. 衝突回避動作生成システム

本システムは、あらかじめロードマップ作成処理と衝突情報取得処理を行っておく事前処理と実時間で動作生成処理を行う本処理によって構成される。これらの処理について次節より述べる。

3.1 ロードマップ作成処理

ロードマップの作成処理では、障害物が存在しない環境において、まず木構造のロードマップを作成し、その後任意の節点間を結ぶことでグラフ構造のロードマップを作成する。これは、分断したロードマップが作成されることを防ぐためである。以下、ロードマップ作成における各処理の詳細について次に述べる。

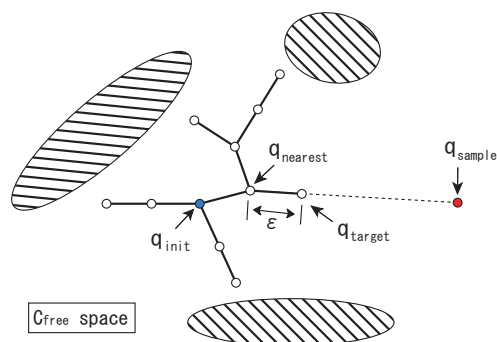


図 2: RRT 法の処理モデル

3.1.1 RRT 法

木構造のロードマップを作成する際には、RRT(Rapidly-exploring Random Tree)法^[7]を用いる。このRRT法は、乱数を用いた経路設計法の一つで、他のアルゴリズムと比較して少ない計算コストで C_{free} 空間内の広範囲にロードマップを作成でき、高次元空間の経路設計に有効であるとされている。

RRT法を用いたロードマップ作成処理のモデルを図2に示す。このように、ある初期点 q_{init} から始め、 C_{free} 空間内のランダムな点 q_{sample} に向かって枝を伸ばして行く処理をくり返すことでロードマップの作成を行う。枝を伸ばす際には、 q_{sample} に最も近いロードマップの節点 $q_{nearest}$ から、長さ ϵ ごとに補間した点 q_{target} を順次作成しながら衝突判定を行う。

このとき、 C 空間内の2点 $q_i - q_j$ 間の距離 D は式1により求められる。

$$D^2 = w_T \|T_i - T_j\|^2 + w_R \sum_n \|R_i^n - R_j^n\|^2 \quad (1)$$

ここで、 w_T, w_R は、それぞれ三次元位置に関する距離要素と関節角度に関する距離要素の重みパラメタである。

3.1.2 ロードマップのグラフ化

作成した木構造のロードマップ内の以下の条件を満たす2節点 $q_i, q_j (i \neq j)$ を接続することで、グラフ構造のロードマップを作成する。

- (1) 2節点間の距離が閾値 TH_{dist} 以下である
- (2) 2節点間に衝突が発生していない

- (3) 2 節点間を結ぶ線分と 2 節点に接続している枝との成す角が閾値 TH_{angle} 以上である

ここで、条件 3 において既に存在する方向の経路が冗長に作成されないように枝の作成を制限している。また、これらの閾値 TH_{dist}, TH_{angle} については実験的に決定しており、今回の実験では $TH_{dist} = 10\varepsilon, TH_{angle} = \pi/4$ としている。

3.1.3 衝突判定

C 空間内の点 q が C_{free} 空間内の点であるかどうかを判定するためには、 q が表すキャラクタの姿勢が衝突を起こしているか、各関節角度が制限範囲内かどうか、の 2 つを判定する。この衝突の判定には ColDet^[8] ライブラリを使用し、これによりキャラクタの各部位と障害物間の衝突判定を行っている。

3.2 衝突情報取得処理

この処理では、キャラクタが動作する三次元仮想空間 W を小さなボクセル集合 V に分割し、各ボクセル $v \in V$ とキャラクタとの衝突判定を行うことで、キャラクタが W 空間内のどの領域と衝突を起こしているかを計算する。これにより、ある姿勢 q のキャラクタが衝突する W 空間内の領域をボクセル集合 $V_q = \{v | \text{isCollided}(q, v) = \text{true}\}$ で表すことができる。ここで、 $\text{isCollided}(q, v)$ はボクセル v を障害物とした場合、姿勢 q のキャラクタがこの障害物と衝突を起こすならば true となるような関数である。この処理をロードマップの節点（つまり姿勢）ごとに行い、各節点が衝突を起こすボクセル集合 V_q を衝突情報として取得しておく。この衝突情報取得処理結果の例を図 3 に示す。赤い立方体が衝突を起こしているボクセルを表している。

3.3 動作生成処理

動作生成処理では入力として、動作の開始姿勢 q_{start} と終了姿勢 q_{goal} 、障害物の情報、さらに事前処理の出力であるロードマップ、衝突情報が与えられる。この動作生成処理の流れは、2.3 節で述べた通りである。このとき、 $N(q_{start}) - N(q_{goal})$ 間のグラフ経路探索には、A* アルゴリズムを用いる。A* アルゴリズムとは、開始点からの経路コストと終了点までの推定コストの和を計算しながら、その総コストが小さい順に未到達経路

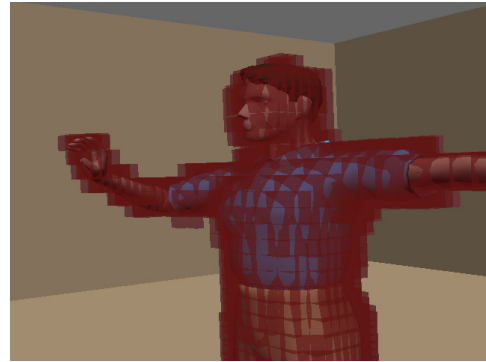


図 3: 衝突情報取得処理結果

の展開を行う経路探索法である。このように終了点までの推定コストを用いることで、開始点から終了点へ向かう経路を優先的に展開することができる。ここで、経路中のある点における経路コストは開始点からの枝の長さの合計、推定コストは終了点までの距離 D によってそれぞれ計算する。

本システムでは、A* アルゴリズムによる経路探索を $N(q_{start}), N(q_{goal})$ の双方向から行う。ここで、障害物の出現・移動・消失によって環境の変化が起こった場合、障害物が存在するボクセルの集合 $V_{obstacle}$ を求める割り込み処理を行う。経路探索時には、各節点において事前処理によって求めた衝突情報 V_q と $V_{obstacle}$ を用いることにより、 $V_q \cap V_{obstacle} \neq \emptyset$ であれば衝突が発生していると高速に判定できる。経路設計の結果、双方向からの経路が同じ節点に到達したとき、これらの 2 つの経路を繋げたものを $N(q_{start}) - N(q_{goal})$ 間の経路として出力する。

4. ロードマップ作成の効率化

4.1 ロードマップ作成の問題点

3 節で述べたように、既存手法^{[1][5]}を組み合わせることによって、動的な環境下でも衝突回避動作生成をある程度行うことが可能である。しかし、以下のような理由からその性能はまだ不十分である。2.4 節で述べたように、安定して衝突回避動作を生成するためには C_{free} 空間全体に密にロードマップを作成する必要がある。一方で、可動部位を増やすなどキャラクタの自由度を高めた動作を生成するためには、 C 空間をより高次

元化する必要がある。 C_{free} 空間全体に密にロードマップを作成するためには、その次元数に対してロードマップの節点数や枝数を指数関数的に増加させる必要があるため、膨大な計算量が必要となってしまふ。また、そのようなロードマップを用いた経路探索処理では計算時間が長くなり、実時間で動作生成が難しくなってしまう。

4.2 コンフィギュレーション空間の低次元化

4.2.1 低次元化の利点

C_{free} 空間によって表現されるキャラクタの姿勢には、たとえ各関節角度が制限角度内であっても人間が普段行わない姿勢が存在する。これは、人間の姿勢には身体構造による制約や動作のパターンなどが存在するためである。つまり、これらの制約を考慮すると C 空間の次元は相関のあるものが多い。そこで、4.1 節で述べた問題点を解決するため、実空間の人間の動作を計測して得たモーションキャプチャデータを用いて C 空間の低次元化を行う。ここでは、このモーションデータの主成分分析によって得られた成分のうち寄与率の低い成分を削減することで、次元の削減を行うこととする。

提案手法では、低次元化によって人間が行う自然な姿勢のみを表現する C' 空間を構成し、この空間内でロードマップの作成を行う。これにより、 C' 空間全体に密なロードマップを作成することで、従来法よりも少ない節点数で動作生成を頑健に行うことができるようになる。

4.2.2 低次元化空間を用いた動作生成

動作生成を行うとき、入力である q_{start}, q_{goal} は C 空間内の点として与えられる。そのため、 C' 空間において作成したロードマップを用いて動作生成を行うためには、 C 空間の点を C' 空間へ射影する必要がある。よって動作生成時には、 q_{start}, q_{goal} を C' 空間の点 q'_{start}, q'_{goal} に射影し、 C' 空間内で経路の設計を行うこととなる。また、得られた経路は削減した次元の値を 0 と考え、逆射影処理を行うことで C 空間における経路を得る。

また、 C' 空間における 2 点間の距離は、それらの点を C 空間に射影し、式 1 を用いることで求めることができる。

4.2.3 モーションキャプチャデータ

本稿では、提案手法の有効性を示すため、腕動作に関するモーションキャプチャデータを用いた C 空間の低次元化を行う。このとき、低次元化に用いるモーションキャプチャデータに動作の偏りがあると、その偏った動作の特徴を大きく反映する C' 空間が構成されてしまい、生成される動作にも偏りが生じてしまう。そのため、様々な特徴を偏りなく保持するモーションキャプチャデータが必要となる。

そこで、本システムでは大西らが行った腕動作の分類⁹⁾に基づくモーションキャプチャデータを用いた。この分類では、人間形ロボットアームが機能的な動作を行いうるような基本動作の提示がなされている。

5. 実験

提案手法の有効性を示すため、次の二種類の実験を行った。また、この実験に用いるロードマップを作成する手法として、次の三つを用いた。

手法 1 確率的ロードマップ法

手法 2 手法 1 でランダム作成する姿勢をモーションデータ内の動作における各関節の稼働角度範囲内に限定した手法

手法 3 提案手法

これらの実験には、CPU Pentium4 3.2GHz、メモリ 1GB の PC を用いた。

5.1 ロードマップ作成実験

5.1.1 実験概要

提案手法によりロードマップが効率的に作成されることを確かめるため、 C_{free} 空間内の点と作成したロードマップとの距離、すなわち、その点に最も近いロードマップの節点との距離を求める。この距離が短いほど、動作生成における開始姿勢や終了姿勢とロードマップ間の接続処理が成功しやすいこととなり、動作生成の失敗が減少する。

本実験では、節点の個数を 1000, 5000, 10000 とし、枝の長さ ε をモーションデータにおける 1/10 秒間隔の姿勢間距離の平均である 0.14 としてロードマップの作成を行った。またロードマッ

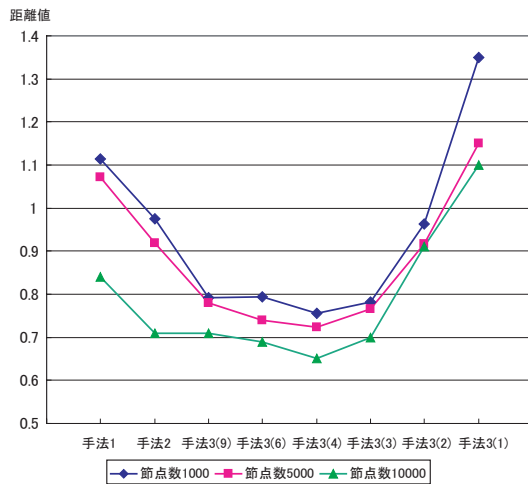


図 4: ロードマップ生成実験結果：括弧内の数字は低次元空間の次元数

ブとの接続を行う姿勢は，低次元化処理に用いるモーションデータとは異なる動作を行うモーションデータ内の姿勢を使用する．このモーションデータ内の任意の 100 個の姿勢と作成したロードマップとの接続距離の平均値を求める．

5.1.2 結果と考察

本実験の結果を図 4 に示す．縦軸に C_{free} 空間内の点とロードマップ間の距離の平均値をとり，三つの折れ線グラフがそれぞれロードマップの節点数を 1000, 5000, 10000 としたときの結果を表している．このように，提案手法を用いることで自然な姿勢が存在する C_{free} 空間内の領域に密なロードマップが作成されていることが確認できる．ただし 1 次元や 2 次元のような C' 空間を用いた場合に，接続距離が長くなってしまっている．これは，次元を削減しすぎたことで， C' 空間において表現できる姿勢が限られてしまったためである．これらの結果から，腕動作の生成には 4 次元の C' 空間を用いたロードマップ作成が妥当であると判断できる．

5.2 動作生成実験

5.2.1 実験概要

図 5,6 のような机，棚と箱型の障害物を配置した実験環境において動作生成実験を行った．本実験では，それぞれ机上，棚内の様々な場所に緑球を配置し，この緑球を掴む右腕動作の生成を行

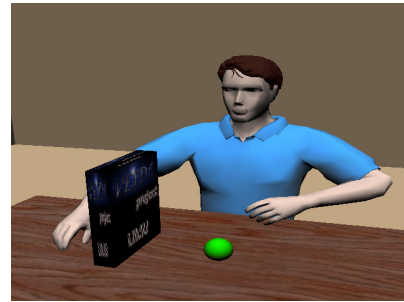


図 5: 机と箱型の障害物を配置した実験環境

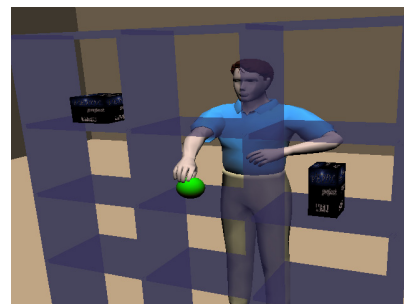


図 6: 棚と箱型の障害物を配置した実験環境

う．ここで，この動作生成には開始姿勢として右腕を下ろしている姿勢，終了姿勢として緑球を右腕で掴むような姿勢を用いる．この緑球を掴む姿勢は，右手先の位置を緑球の位置として逆運動学法を用いて決定する．

本実験は手法 1 により作成したロードマップと，手法 3 により低次元化した 4 次元 C' 空間において作成したロードマップを用いて行う．このとき，緑球の位置を変化させながら動作生成を 100 回繰り返し行い，この動作生成の成功回数を調べる．ここで，開始姿勢や終了姿勢とロードマップとの接続が失敗した場合，ロードマップの分断によって経路探索が失敗した場合を動作生成の失敗とし，これらの処理が成功した場合を動作生成の成功とする．

5.2.2 結果と考察

節点数 5000 のロードマップを用いて行った動作生成の成功・失敗回数を表 1 に示す．ここで，第 3,4 項はそれぞれ動作生成が失敗した場合の中で，開始姿勢や終了姿勢とロードマップ間の接続に失敗した場合と，経路探索に失敗した場合の

表 1: 動作生成の成功・失敗回数

	生成 成功	生成失敗		計
		接続失敗	探索失敗	
手法 1(図 4)	68	17	15	100
手法 3(図 4)	82	12	6	100
手法 1(図 5)	49	28	23	100
手法 3(図 5)	68	20	12	100

表 2: ロードマップの節点数ごとの動作生成時間

節点数	1000	5000	10000
処理時間 [sec]	0.297	0.503	1.01

回数である。

このように、提案手法を用いた場合に動作生成の成功回数が増加していることがわかる。特に提案手法によってロードマップが密に作成されたことにより、ロードマップの分断による動作生成の失敗が大きく減少している。節点数 1000, 10000 のロードマップを用いた場合でも同様の傾向が見られた。

また、表 2 に本実験におけるロードマップの節点数ごとの動作生成の処理平均時間を示す。このように、節点数の増加によって接続処理回数や経路数が増加し、動作生成に必要な処理時間が増加してしまう。そのため、提案手法によって節点数を削減することができ、動作生成を高速に行うことができることになる。

6. おわりに

本稿では、コンフィギュレーション空間の低次元化によりロードマップを効率的に作成する手法を提案した。また、従来の確率的ロードマップ法を用いた動作生成との比較実験を行うことで、提案手法の有効性を示した。今後の課題と展開として以下のことが挙げられる。

全身動作への拡張 提案手法を用いてキャラクタ全身の衝突回避動作の生成が行えるようにシステムを拡張する。

キャラクタ同士の衝突回避 2 体以上のキャラクタが同一空間内で動作するとき、これらのキャラクタ間の衝突回避を考慮した動作生成を行う。

動作の平滑化 得られた経路のショートカット処理や平滑化処理を行うことで、動作をより滑らかなものとする。

謝辞

本研究の一部は、科学研究費補助金若手研究(B)「リアルタイムヒューマンプロキシを用いたアバターベース遠隔地コミュニケーションの研究(16700108)」の補助を受けた。

参考文献

- [1] L. Kavraki, P. Svestka, J. Latombe, M. Overmars, "Probabilistic Roadmaps for Fast Path Planning in High Dimensional Configuration Spaces," IEEE Transactions on Robotics and Automation, vol. 12, pp.566-580, 1996.
- [2] Y. Liu, N. Badler, "Real-time reach planning for animated characters using hardware acceleration," Proc. of the International Conference on Computer Animation and Social Agents, Philadelphia, USA, pp.86-93, 2003.
- [3] J.J. Kuffner, J.C. Latombe, "Interactive manipulation planning for animated characters," In Pacific Graphics '00, Hong Kong, pp.417-418, 2000.
- [4] M. Kallmann, A. Aubel, T. Abaci, D. Thalmann, "Planning Collision-Free Reaching Motions for Interactive Object Manipulation and Grasping," Proc. of Eurographics, Granada, Spain, pp.313-322, 2003.
- [5] M. Kallmann, M. Mataric, "Motion Planning Using Dynamic Roadmaps," Proc. of the IEEE International Conference on Robotics and Automation, New Orleans, Louisiana, pp.4399-4404, 2004.
- [6] R. Geraerts, M. Overmars, "A Comparative Study of Probabilistic Roadmap Planners," Proc. of the International Workshop on Algorithmic Foundations of Robotics, Nice, France, 2002.
- [7] S. M. LaValle, "Rapidly-exploring random trees: a new tool for path planning," Technical Report No.98-11, Computer Science Dept, Iowa State University, 1998.
- [8] ColDet - Free 3D Collision Detection Library : <http://coldet.sourceforge.net/>
- [9] 大西謙吾, 宮川浩臣, 田島孝光, 斎藤之男, "人間形ロボットハンドの高機能化に関する研究 -ハンド・アームの動作分類を用いた接触覚センサベース制御-", パイオメカニズム, 16, pp.155-165, 2002.