動的ロードマップによるキャラクタ動作生成の高速化 : コンフィギュレーション空間の低次元化処理

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
財団法人九州システム情報技術研究所

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

出版情報：電子情報通信学会技術研究報告. MVE, マルチメディア・仮想環境基礎. 106 (611), pp.29-34, 2007-03-16. 電子情報通信学会
バージョン：
権利関係：

社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
MVE2006-83(2007-3)

動的ロードマップによるキャラクタ動作生成の高速化

コンフィギュレーション空間の低次元化処理

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 〒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
††† 九州大学大学院システム情報科学研究院 〒819–0395 福岡県福岡市西区元岡 744
E-mail: †{nomiyama,rin}@limu.is.kyushu-u.ac.jp, ††arita@isit.or.jp

あらまし 著者らは，仮想環境中のキャラクタが行う衝突回避動作の実時間生成について研究を行っている．既存の衝突回避動作の生成手法では，膨大な事前情報を用いることで高速な動作生成を行う．そのため，生成動作の多様化により動作生成時間が増加してしまうという欠点を持つ．そこで本稿では，モーションキャプチャデータの主成分を用いたコンフィギュレーション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，事前に取得すべき情報の削減を行うことができる．この提案手法によって高速で頑健な衝突回避動作生成が行えることを，従来の確率的ロードマップ法による動作生成との比較実験を行うことにより示す．

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

Fast Motion Planning Using Dynamic Roadmaps

Dimensional Reduction of the Configuration Space

Hideto NOMIYAMA†, Daisaku ARITA††, and 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
††† Department of Intelligent Systems, Kyushu University 744, Motooka, Nishi-ku, Fukuoka, 819–0395, Japan
E-mail: †{nomiyama,rin}@limu.is.kyushu-u.ac.jp, ††arita@isit.or.jp

Abstract Our research is aiming at real-time collision-free motion planning for character animation in virtual environments. Our system performs a fast motion planning using precomputed Dynamic Roadmaps. However, there is a problem that the processing time of motion planning is too long for real-time character animation since robust planning of various motions requires an enormous size of precomputed Dynamic Roadmaps. To solve this problem, we propose a new method using a dimensionally reduced configuration space, which includes only natural postures of a character, by analyzing principal components of a motion capture data. Our method allows fast and robust motion planning in dynamic environments. 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，インタラクションやコミュニケーションにおいて仮想環境が用いられる場が増えてきた．これは，仮想環境を用いることで容易に様々な場面を再現することが可能なためである．この背景に伴い，人間を模した CG キャラクタに様々な動作を自然に行わせるための研究が盛んに行われている．特に，衝突回避動作の生

成は環境の変化に応じて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．衝突回避動作の生成では，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．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，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ように，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．一方，multiple-query では，高速な動作経路設計を行うことができるように，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．これにより，幾つかの動作をロードマップのグラフ探索を用いて生成することができる．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]〜[5].

インタラクションシステムやコミュニケーションシステムでは実時間処理が不可欠であり，円滑にキャラクタを動作させる必要がある．そのために，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．しかし，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ってしまい，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生成する必要性は高い．

本研究では，実時間処理という観点から multiple-query に基づく既存手法である動的ロードマップを用いた実時間衝突回避動作の生成を行う．ただし，動的ロードマップにより多様な動作を生成するためには，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．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を用いた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．

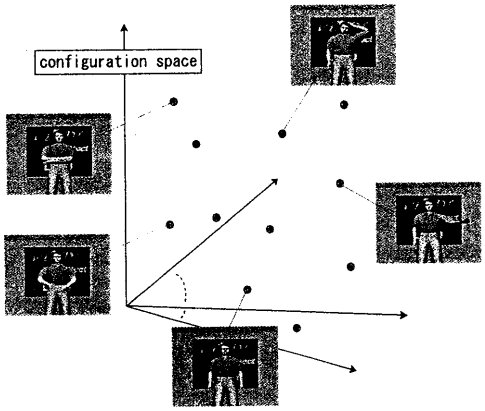

図 1　コンフィギュレーション空間

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 $(T, R^{head}, R^{righthand}, R^{lefthand}, \cdots)$ 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 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} 空間のような多次元空間において経路を作成する場合，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．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 $\{q_1, q_2, q_3, \cdots\}$ 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．

ロードマップ作成については，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，それぞれ q_{start}, q_{goal}

とロードマップの節点間に衝突が発生していない線分を見つけ，そのときの節点を $N(q_{start}), N(q_{goal})$ とする．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 $q_{start} \rightarrow P \rightarrow q_{goal}$ が衝突回避動作として出力される．

2.4 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，single-query では経路設計に多くの時間がかかるという欠点がある．

これに対し multiple-query 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二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，幾つかの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 $N(q_{start}), N(q_{goal})$ を見つけるために衝突判定処理が必要となる．さらに場合によっては $N(q_{start}), N(q_{goal})$ が見つからず，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．このため本研究では，確率的ロードマップ法に Kallmann らが提案するロードマップ更新法 [5] を加えた動的ロードマップを用いて動作の生成を行う．

3. 動的ロードマップを用いた衝突回避動作生成

動的ロードマップは，事前処理であるロードマップ作成処理と衝突情報取得処理により作成される．本処理では，動的ロードマップを用いて実時間で動作生成処理を行う．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，その後適当な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，事前処理の段階で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．

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

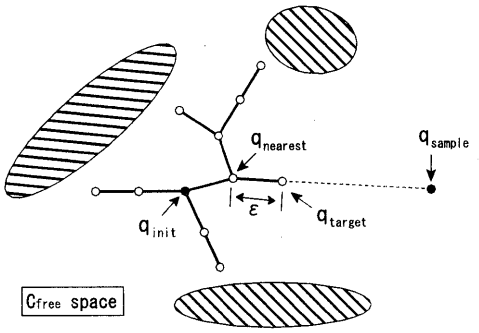

図 2　RRT 法の処理モデル

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．

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

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．

また，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 $V_q = \{v | \text{isCollided}(q, v) = true\}$ 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動的ロードマップが与えられる．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

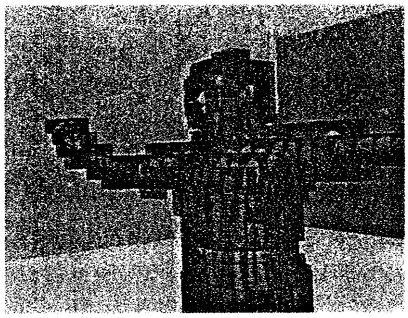

図 3　衝突情報取得処理結果

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．

また障害物の出現・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．

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といえる．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う．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，実時間での動作生成が難しくなってしまう．

4.2 問題解決へのアプローチ

実用性のある計算量で安定した動作生成を行うためのアプローチとして，ロードマップの作成をある限定された範囲内において行う手法が一般に用いられる．具体的に既存の動作生成法では，生成する動作に応じてキャラクタの各関節の稼働範囲を制限したり [4]，部位位置の比較により類似した姿勢を制限する [5]，といった処理が行われている．しかし，このような処理では，制限範囲の実験的な調整が必要であり，さらに生成可能な動作が限られてしまう．

そこで提案手法では，実世界の人間の動作に着目することで動作生成に冗長な姿勢を削減する．これにより，生成可能な動作を限定すること無く，計算量の削減を行うことができる．また，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ることで，パラメタ調整をモーションデータという統計的な情報により自動的に行うことができる．この提案手法について，次節にその詳細を述べる．

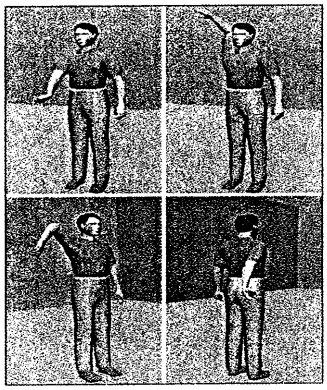

図 4　人間が普段行わない姿勢の例

4.3 コンフィギュレーション空間の低次元化

4.3.1 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（図 4）．これは，人間の姿勢には身体構造による部位間の依存関係や頻出する動作パタンなどが存在するためである．つまり，これらの性質を考慮すると C 空間の成分には相関のあるものが多い．そこで，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．

本手法では，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．C' 空間全体に密なロードマップを作成することで，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．

4.3.2 C' 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 q を C' 空間へ射影し，点 q' を得る処理が必要となる．この射影処理は，C 空間を n 次元空間，C' 空間を m 次元空間としたとき，ある $m \times n$ の射影行列 $\mathbf{M}$ とモーションデータの平均姿勢 $\bar{q}$ を用いて式 2 により計算される．

$$q' = f(q) = \mathbf{M}(q - \bar{q}) \tag{2}$$

また，C' 空間から C 空間への逆射影処理は，$\mathbf{M}$ の擬似逆行列 $\mathbf{M}^+$ を用いて式 3 により計算される．

$$q = f^{-1}(q') = \bar{q} + \mathbf{M}^+ q' \tag{3}$$

この射影・逆斜影処理を加えた動作生成処理は以下の流れで行われる．

（1） C 空間の点 q_{start}, q_{goal} を C' 空間に射影し，q'_{start}, q'_{goal} を得る

（2） q'_{start}, q'_{goal} と接続するロードマップの節点 $N(q'_{start}), N(q'_{goal})$ を見つける

（3） ロードマップ上で $N(q'_{start}) - N(q'_{goal})$ 間のグラフ経路探索を行う

（4） 経路 $P' = (q'_{start} \rightarrow N(q'_{start}) \rightarrow \cdots \rightarrow N(q'_{goal}) \rightarrow q'_{goal})$ を得る

（5） 経路 P' の点を C 空間に写像し，$P = (f^{-1}(q'_{start}) \rightarrow f^{-1}(N(q'_{start})) \rightarrow \cdots \rightarrow f^{-1}(N(q'_{goal})) \rightarrow f^{-1}(q'_{goal})) = (q_{start} \rightarrow f^{-1}(N(q'_{start})) \rightarrow \cdots \rightarrow f^{-1}(N(q'_{goal})) \rightarrow q_{goal})$ を得る

（6） 経路 P を出力する

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

4.3.3 モーションキャプチャデータ

本稿では，本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] を満遍なく行うような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，二種類の実験を行う．このとき，この実験に用いるロードマップを作成する手法として，次の三つを用いる．

- 提案手法（手法 1）
- 確率的ロードマップ法（手法 2）
- ロードマップ作成の標本点として用い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動角度範囲内に限定した手法（手法 3）

手法 3 では，モーションデータを基にして各関節角度の制限を行うことでロードマップの作成する範囲を限定し，ロードマップ作成の効率化を行っている．ただし，4.3.1 節で述べた C 空間の次元の相関は考慮されていない．

また，これらの実験には，CPU Pentium4 3.2GHz，メモリ 1GB の PC を用いる．

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，C 空間に対して節点数の少ないロードマップを用いた動作生成を想定して，節点の個数を 1000，5000，10000 として各ロードマップの作成を行う．さらに，ロードマップの枝の長さ 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う．またロードマップ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．

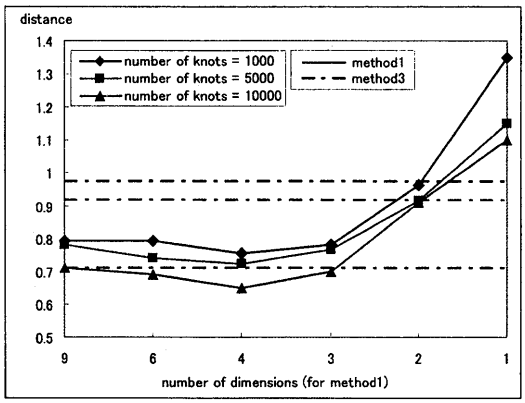

図 5　ロードマップ生成実験結果：手法 1 と手法 3 の比較

5.1.2 結果と考察

手法 1 と手法 3 の比較を図 5 に示す．縦軸に接続距離の平均値をとり，ロードマップの節点数を 10000，5000，1000 としたときの結果をそれぞれ，手法 1 を折れ線で，手法 3 を破線で表している．このように，同じ節点数のロードマップを作成した際，提案手法を用いることで C_{free} 空間内の自然な姿勢が存在する領域により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，次元が削減されるにつれて，C' 空間において表現できる姿勢が限られていってしまうためである．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 6，図 7 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う．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．

本実験は提案手法（手法 1）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と，確率的ロードマップ法（手法 2）により作成したロードマップをそれぞれ節点数を 5000 として事前に作成しておき，これらを用いて動作の生成を行う．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

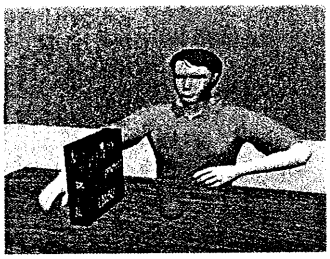

図 6　机と箱型の障害物を配置した実験環境

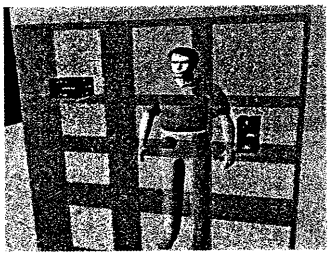

図 7　棚と箱型の障害物を配置した実験環境

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

動作生成の成功・失敗回数を表 1 に示す. ここで, 第三項, 第四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回数である. 提案手法を用いた場合に接続失敗・探索失敗回数がそれぞれ減少し, 動作生成の成功回数が増加している. この結果より, 提案手法によってロードマップが効率的に作成され, 動作生成がより頑健に行えるようになったことがわかる.

このとき, 動作経路として出力された平均節点数は, 従来法を用いた場合では 50.4, 提案手法を用いた場合では 44.3 であった. これは, 提案手法ではロードマップを密に作成したことにより, 短い動作経路で衝突回避が実現できていることを示している.

さらに, 表 2 に図 6 の環境を用いた動作生成実験におけるロードマップの節点数ごとの動作生成処理の平均時間を示す. このように, 提案手法を用いることで動作生成の処理時間が短縮できていることがわかる. これは, ロードマップを効率的に作成することで接続距離が短くなり, 接続処理に必要となる衝突判定の回数を減らすことができたためである. また, 節点数が増加するにつれて処理時間も増加していることがわかる. これは, 節点の数だけ行う必要のある各計算処理の回数や探索する経路数が増加してしまうためである. よって, 提案手法によって動作生成に必要なロードマップの節点数を削減することも, 動作生成の高速化に繋がっていくこととなる.

表 1　動作生成の成功・失敗回数

	generation success	generation failure		sum
		connection	planning	
method1(figure6)	82	12	6	100
method2(figure6)	68	17	15	100
method1(figure7)	68	20	12	100
method2(figure7)	49	28	23	100

表 2　動作生成処理の平均時間 [sec]

	knot num		
	1000	5000	10000
method1	0.4876	0.6742	0.9795
method2	0.7345	0.8433	1.1316

6. おわりに

本稿では, 高速で頑健な衝突回避動作生成を行うための手法として, コンフィギュレーション空間の低次元化を提案した. 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.

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の関する研究 –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–", バイオメカニズム, 16, pp.155-165, 2002.