Information Processing Society of Japan

社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2005－CG－121 (10)
2005／11／18

幾何制約解消による3次元事象作図支援

柏原 将輝† 白井 靖人‡

概要

3次元事象の作図時に、3次元CGソフトウェアを使用する場合、スクリーンを通して仮想的な3次元空間中に作図するため、作図に慣れていないユーザでは、3次元事象を構成する幾何学図形間の位置関係を正確に維持しながら作図することは困難であると考えられる。本研究では、この位置関係に焦点を当て、正確に3次元事象を作図するための手法を提案する。本手法では、包含、平行、垂直の位置関係を対象とし、それらを幾何制約で記述し、解消することで位置関係を維持する。幾何制約は、作図者が、対象とする幾何学図形と位置関係を明示的に選択することで追加される。実際に、本手法を採用した作図ツールを開発することにより、その支援効果を評価する。

Drawing support of 3-dimensional phenomenon by geometric constraint satisfaction

Masateru Kashiwabara† Yasuto Shirai‡

ABSTRACT

When users who aren't accustomed to drawing figures model a 3-dimensional phenomenon with 3-dimensional CG software, they are difficult to draw a figure with the positions of geometric configurations in a 3-dimensional phenomenon because of drawing figures in a virtual 3-dimensional space with 2D screen. We focus on the positions of them and propose a method with which everyone can draw a 3-dimensional phenomenon correctly. This method keeps the positions of geometric configurations by describing and satisfying three positions: inclusion, parallel, and vertical with constraints. A constraint is added when users select two intended geometric configurations and a position of them. We will develop a drawing tool into which this method is built and evaluate its effect of support.

1. はじめに

実空間における事象、つまり、3次元事象を説明する場合に図を用いることがある。一般に、こういった図は、3次元事象に含まれる事物を抽象化した幾何学図形によって構成され、その意味を表すラベルとともに表現される。このような事象を図示する場合、ノートや黒板などの2次元のキャンバス上に作図する方法と3次元CGソフトウェアなどを利用することにより、仮想的な3次元空間に作図する方法が考えられる。

前者の場合は、空間から平面へ置き換えるため、3次元事象が持つ意味を全て表現することが難しく、図を見る側の理解が作図者の描き方に大きく依存すると考えられる。一方、後者の場合は、空間のまま表現するため、正確に作図することができれば、誰でも同じ3次元事象を描くことができる。しかし、作図者は、スクリーンを通して仮想的な3次元空間中に図を描いていくため、作図に慣れていないユーザでは、正確に作図することは困難であると考えられる。

そこで、本研究では、正確に3次元事象を作図するための手法を提案し、本手法を採用した作図ツールを開発することにより、その支援効果を評価する。

図1は、デザルクの定理を表したものだが、この図を構成する幾何学図形間には幾つかの位置関係が存在する。例えば、点Aは平面α上に位置し、点Bは点A

†静岡大学院情報学研究科
Graduate School of Information, Shizuoka University
‡静岡大学情報学部
Faculty of Information, Shizuoka University

NII-Electronic Library Service

Information Processing Society of Japan

と2平面の交線上の1点を結ぶ線分上に位置する。このような位置関係は正確に3次元事象を作図する上で重要な要素であり、本手法では、この位置関係を幾何制約[1][2]で記述し、それを解消することで位置関係を維持する。具体的には、作図者が位置関係と対象とする幾何学図形を指定すると、システムは与えられた幾何制約を解消し、指定された幾何学図形間の位置関係を維持する。その結果、作図者は幾何学図形間の位置関係を意識することなく、正確に作図することができる。

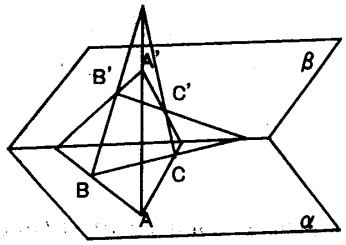


図1：デザルクの定理

2. 幾何制約

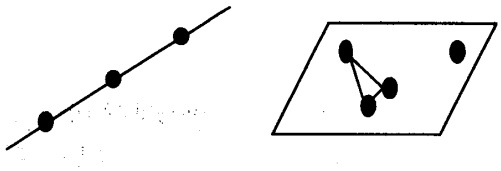
研究に先立ち、幾何学[3]の観点から、様々な分野における3次元事象を示す概念的な図を調べた。その結果、包含関係、平行関係、垂直関係の3つの位置関係が、多くの3次元事象で見られた。本研究では、これら3つの位置関係を対象とする。

2.1 包含制約

点と直線(または、半直線、線分)、点と平面の包含関係を維持するための幾何制約である。図2の(a)は点と直線の包含関係を示しており、この図は、ある点がある直線上に位置する、もしくは、ある直線がある点を通過するという位置関係を表している。また、図2の(b)は、点と平面の包含関係を示しており、この図は、ある点がある平面上に位置する、もしくは、ある平面がある点を通過するという位置関係を表している。これらの位置関係を包含制約で記述し、解消することにより、例えば、図2の(a)において、ある直線がある点を通過するという位置関係である場合、点を動かすと、対応する直線がその点を通過するように移動する。また、直線の代わりに半直線、線分を扱う場合では、点の位置は線分、半直線上の端点内に制限される。

2.2 平行制約

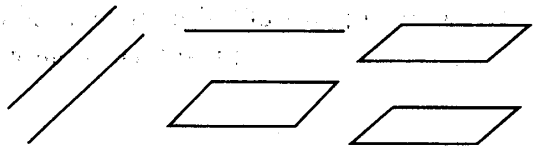
直線と直線、直線と平面、平面と平面の平行関係を



(a)点と直線　　(b)点と平面

図2：包含関係

維持するための制約である。図3の(a)は、2直線の平行関係を示しており、図3の(b)は、直線と平面の平行関係を示している。また、図3の(c)は、2平面の平行関係を示している。これらの位置関係を平行制約で記述し、解消することにより、例えば、図3の(b)の場合では、直線、もしくは、平面を動かすと、対応する直線、もしくは平面がそれと平行な位置に移動する。

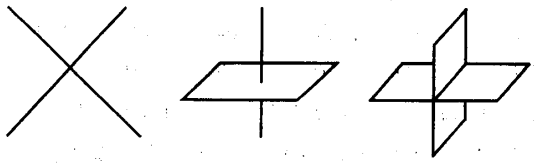


(a)直線と直線　(b)直線と平面　(c)平面と平面

図3：平行関係

2.3 垂直制約

直線と直線、直線と平面、平面と平面の垂直関係を維持するための制約である。図4の(a)は、2直線の垂直関係を示しており、図4の(b)は、直線と平面の垂直関係を示している。また、図4の(c)は、2平面の垂直関係を示している。これらの関係を垂直制約で記述し、解消することにより、例えば、図4の(c)の場合では、一方の平面を動かすと、対応するもう一方の平面がそれと垂直な位置に移動する。



(a)直線と直線　(b)直線と平面　(c)平面と平面

図4：垂直関係

3. データ構造

ここでは、2章で述べた幾何制約を定義するためのデータ構造について述べる。本手法で扱うデータ構造

NII-Electronic Library Service

Information Processing Society of Japan

は、一般に、3 次元グラフィクスでよく用いられる、シーングラフを拡張したものとする。

3.1 シーングラフ

シーングラフ[4]とは、3 次元仮想空間を概念的に表現するための構造である DAG(Directed Acyclic Graph)を用いて表現される。通常、シーングラフは、ルートノード、形状ノード、グループノードで構成され、図 5 のような形で表現される。

図 5 の R はルートノード、S は形状ノード、G はグループノードを表す。本研究では、先行研究[5]と同様で、これらに加えて、制約ノード(図 5 の C)を追加することで、シーングラフに幾何制約を導入する。

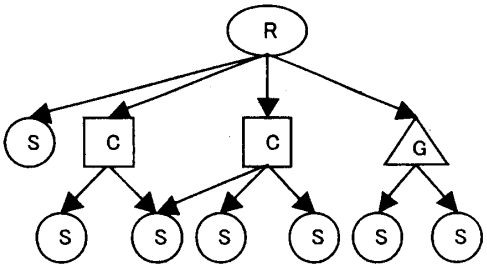


図 5：シーングラフ

3.2 ルートノード

世界座標系での仮想的な根を表すノードであり、このノードから他の全てのノードへ辿ることができる。また、実際にスクリーン上へ投影するための仮想的なスクリーンを定義するカメラを内部で保持する。

3.3 形状ノード

3 次元事象を構成する幾何学図形を保持するノードである。本研究で扱う幾何学図形は、点、線分、閉じた折れ線、直線、半直線、平面の 6 種類であり、これらを組み合わせることによって 3 次元事象が構築される。ここで、線分、直線、半直線は 2 つの点を指定することにより定義され、平面は、3 つの点を指定することにより定義される。また、閉じた折れ線は、3 つ以上の点を指定することにより定義される。

3.4 グループノード

複数のノードをグループ化するノードである。グループノードは、複数のノードを指定することにより定義され、それらを代表するノードとなる。指定可能なノードは、形状ノード、制約ノード、グループノードのいずれかである。

3.5 制約ノード

幾何学図形間の位置関係を維持するための幾何制約を保持するノードである。ここで、保持される幾何制約は、2 章で述べた包含制約、平行制約、垂直制約のいずれかである。図 6 は、平面 α と 3 つの点 A, B, C に対して包含制約を適用させた場合のシーングラフである。また、本手法で扱う幾何制約では、幾何学図形間で一定の方向が存在し、その方向も合わせて保持される。

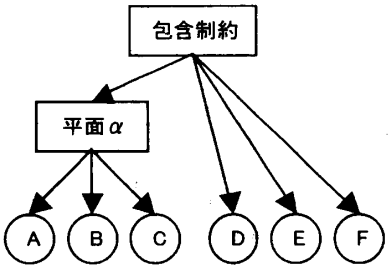


図 6：包含制約

4. 幾何制約の適用

ここでは、任意の幾何学図形に対して幾何制約を適用するための方法について述べる。本手法では、対象とする幾何学図形とそれらの位置関係をユーザが明示的に宣言することにより、任意の幾何学図形に対して幾何制約を適用する。しかし、その場合、同じ幾何学図形に対して複数の幾何制約が適用される可能性があるため、本手法では、図 7 のように、幾何制約を追加する前処理として、衝突、循環の検出を行い、それぞれにおいて異なる対処法を適用する。衝突、循環の検出、及び対処法については次に述べる。

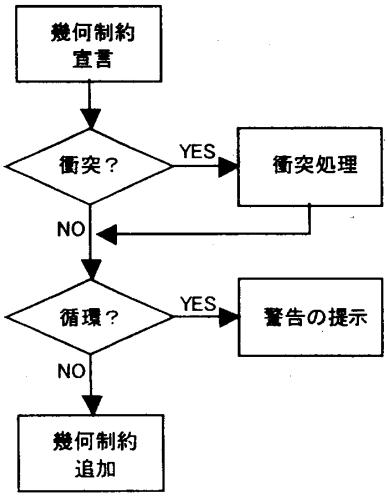


図 7：幾何制約の追加

NII-Electronic Library Service

Information Processing Society of Japan

4.1 衝突の検出、及び対処法

衝突とは、1つの幾何学図形が複数の幾何制約の影響を受ける場合を意味する。本手法では、衝突の検出を行うために、表1のように、幾何制約(Constraint)、制約適用元(From)、制約適用先(To)からなる制約テーブルを用意する。ここで、制約適用元とは、幾何制約のパラメータとなる幾何学図形のことであり、この幾何学図形の位置が、制約適用先の幾何学図形の位置に対して影響を与え、位置関係を維持する。

表1：制約テーブル

Constraint	From	To
包含制約 0	平面(A, B, C)	D
包含制約 1	直線(E, F)	A
平行制約 0	直線(E, F)	直線(G, H)
包含制約 2	直線(E, F)	D
包含制約 3	直線(D, E)	A

本手法では、制約テーブルの制約適用先に着目し、同じ幾何学図形が含まれているか調べることによって、衝突の検出を行う。そして、同じ幾何学図形が含まれていた場合、衝突が起こると考える。表1には、包含制約0と包含制約2の制約適用先で点Dが含まれているため、衝突が起こる。

衝突を回避するための対処法としては、幾何制約を一つにまとめるという方法をとる。例えば、表1のように点Dにおいて衝突が起こる場合、関連する2つの包含制約を一つにまとめることにより、衝突を回避することができる。衝突回避後の表1の制約テーブルは、表2のようになる。

表2：衝突回避後の制約テーブル

Constraint	From	To
包含制約 0	平面(A, B, C), 直線(E, F)	D
包含制約 1	直線(E, F)	A
平行制約 0	直線(E, F)	直線(G, H)
平行制約 3	直線(D, G)	A

4.2 循環の検出、及び対処法

循環とは、制約適用元の幾何学図形が制約適用先の幾何学図形に影響を受ける場合を意味する。本手法では、循環を検出するために、制約テーブルを元に、表3のように、点(Point)とその点が影響を与える点(Affected-Point)からなる循環検出テーブルを作成する。制約テーブルのデータは、追加された順に参照され、以下の順序で生成される。

1. 平面や直線を定義する点も含めた、制約適用元に含まれる点を1つずつ循環検出テーブルのPointへ追加する。ここで、その点がすでに追加済みの点であれば追加しない。
2. 平面や直線を定義する点も含めた、制約適用先に含まれる全ての点を 1 で追加した点に対応するAffected-Pointへ追加する。ここで、1で追加した点が、他の Point に対応する Affected-Point であった場合、そのPointにも点を追加する。
3. 制約テーブルのデータが存在すれば、参照するデータを一つ進め、1へ。存在しなければ終了。

表3：循環検出テーブル

Point	Affected-Point
A	D, A
B	D
C	D
E	D, A
F	D, A
D	D, A, H
G	A

本手法では、点とその点が影響を与える点に同じ点が含まれているか比較することで循環の検出を行う。そして、同じ点が含まれていた場合、循環が起こると考える。表5では、点Aが影響を与える点に点Aが含まれているため、循環が起こる。

本手法における幾何制約は一定の方向を持つ単方向制約であるため、循環が起こる場合は、ユーザ側へ関連する幾何制約とともに警告を提示することで対処する。

5. 制約解消

ここでは、本手法における幾何制約の解消法について述べていく。2章で述べた包含制約、平行制約、垂直制約の解消の流れ、及び、複数の幾何制約に対する処理について次で示す。

5.1 包含制約の解消

包含制約の解消の例として、ある点がある平面上に位置する場合の制約解消の流れを図8に示す。

① 平面を構成する3つの点をパラメータとして平面の式を生成し、制約ノードで保持する。この場合、平面の式が幾何制約となる。

NII-Electronic Library Service

Information Processing Society of Japan

② 制約ノードで対応する点を受け取り、その点が線形等式を満たすように制約解消を行い、その結果を返す。

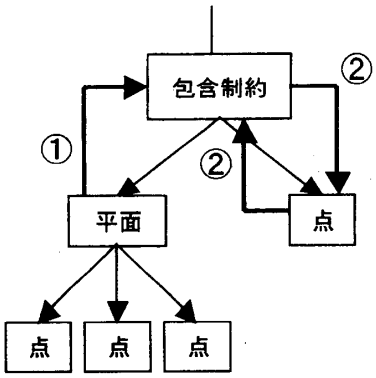


図 8：包含制約の解消

5.2 平行制約の解消

平行制約の解消の例として、2 直線が平行関係を維持する場合の解消の流れを図 9 に示す。

① 直線 1 を構成する 2 つの点と、直線 2 を構成する点の 1 つから、直線 2 の 1 点を通る直線の式を生成し、制約ノードで保持する。この場合、直線の式が幾何制約となる。

② 制約ノードで、直線 2 を構成する残りの点を受け取り、その点が線形等式を満たすように制約解消を行い、その結果を返す。

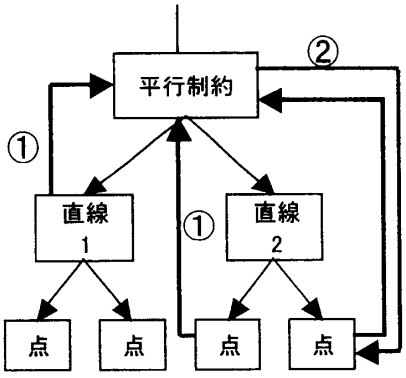


図 9：平行制約の解消

5.3 垂直制約の解消

垂直制約の解消の例として、2 平面が垂直関係を維持する場合の解消の流れを図 10 に示す。

① 平面 1 を構成する 3 つの点と、平面 2 を構成する点の 1 つから、平面 2 の 1 点を通る平面の式を生成し、制約ノードで保持する。この場合、平面の式が幾何制約となる。

② 制約ノードで、平面 2 を構成する残りの 2 つの点を受け取り、それぞれの点が線形等式を満たすように制約解消を行い、その結果を返す。

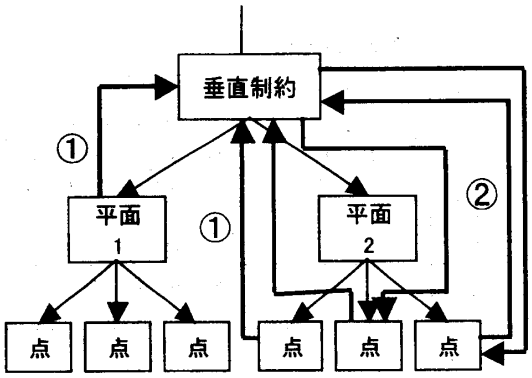


図 10：垂直制約の解消

5.4 複数の幾何制約に対する処理

幾何制約が複数存在する場合、単体での制約解消を行う前に、幾何制約の絞込みと制約解消の順序を決定する必要がある。例えば、表 4 のように幾何制約が追加された場合、上から順に幾何制約を解消していくと、包含制約 1 を解消後、平面(E, F, H)が、平行制約 0 によって変化してしまい、もう一度、包含制約 1 を解消しなければならないことになる。また、点 D を修正した場合、関連する幾何制約は包含制約 1 のみであり、他の幾何制約の解消は必要ない。

表 4：追加された幾何制約

Constraint	From	To
包含制約 0	平面(A, B, C)	D
包含制約 1	平面(E, F, H)	G
平行制約 0	平面(A, B, C)	直線(E, F, H)

そこで、修正した点に応じて、関連する幾何制約のみを絞り込むという処理を行う。表 5 は、表 4 から作成した循環検出テーブルに点ごとで関連する幾何制約を追加したものである。この表から、関連する幾何制約を絞り込む。例えば、点 D を修正した場合、関連する幾何制約は包含制約 0 のみであることが分かる。また、点 A を修正した場合は、まず、平行制約 0, 包含制約 0 が必要であることが分かり、点 A の変化で影響を受ける点 E を見て、さらに包含制約 1 も必要であることが分かり、結果、全ての幾何制約の解消が必要であることが分かる。

NII-Electronic Library Service

Information Processing Society of Japan

続いて、絞り込んだ幾何制約を解消していく順序を決定する。本手法では、トポロジカルソート[6]を適用し、幾何制約を直列に並び替え、その後、それぞれの幾何制約において解消を行う。

表 5：制約絞込みテーブル

Point	Affected	Constraint
A	D, E, F, H	平行制約0, 包含制約0
B	D, E, F, H	平行制約0, 包含制約0
C	D, E, F, H	平行制約0, 包含制約0
D	-	包含制約0
E	G	平行制約0, 包含制約1
F	G	平行制約0, 包含制約1
G	G	包含制約1
H	-	平行制約0, 包含制約1

6. 実装

本研究では、本提案手法を実際に組み込んだ作図ツールを開発することで、その支援効果を評価する。本研究の作図ツールは Java アプリケーションとして実装を行った。実装した作図ツールは図 11 のように、上面図、正面図、側面図の三面図に加えて、作図途中の 3 次元事象を確認するための自由に視点を変更が可能なカメラ図も追加した。現段階では、シーングラフ、ルート、形状、グループノード、作図における基本機能（拡大縮小, Undo, 原点位置の変更など）については実装を完了しているが、幾何制約の部分については未実装である。

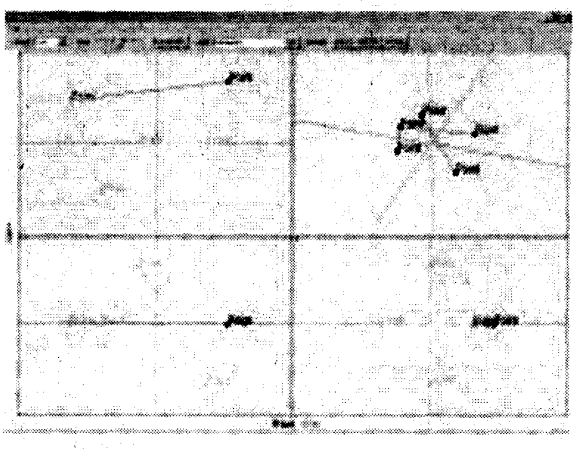


図 11：作図ツール

7. 考察

前章で述べたように、現在、幾何制約についてはまったく未実装である。今後、これまでに述べた方法で、作図ツール上に幾何制約の部分を実装していく予定であるが、幾何制約の解消によって変化する幾何学図形は、修正を行っているキャンバスに着目し、ユーザにとって、なるべく違和感のないように解消を行うようにし、また、現在、存在する幾何制約をオプションとして表示させることで、ユーザに幾何学図形間の位置関係を再認識できるようにする予定である。

幾何制約の部分の実装後、幾つかの 3 次元事象を作図することにより、作図支援としての評価を行う。また、本手法では扱う幾何制約として、包含制約、平行制約、垂直制約のみを扱ったが、必要であれば、距離や角度に関する幾何制約についても検討を行っていく。

参考文献

[1] 金原史和, 佐藤真一, 濱田喬：図形間の幾何的および概念的関係を用いた作図支援, 情報処理学会論文誌, Vol.35, No.5, pp.897-907, (1994)

[2] 細部博史：対話型 3 次元アプリケーションのための幾何制約解消法, 情報処理学会論文誌, Vol.44, No.2, pp.486-495, (2003)

[3] 栗田稔：「立体幾何」数学ワンポイント双書 30, 共立出版, (1985)

[4] 青野雅樹：Java で学ぶコンピュータグラフィックス, オーム社, (2002)

[5] 山田真也, "3 次元事象を可視化するための作図ツールの開発, グラフィクスと CAD 研究報告, Vol.2004, No.121, pp.127-132, (2004)

[6] 石畑清：アルゴリズムとデータ構造, 岩波書店, (1989)

NII-Electronic Library Service