

三次元NCデータ編集ソフトの開発に関する研究*

外山 真也^{*1}・佐藤 征亜^{*1}

Development of Editor for 3-Dimension NC Data

Masaya TOYAMA and Masatsugu SATOH

三次元CAD/CAMシステムを利用して作成したNCデータの容量は、一般的に500 KB (約25000行程度) を超えるデータとなることが多い。このようなNCデータの修正編集は、修正箇所の探索だけでも容易ではなく、かつNCデータを直接編集しようとする、相対指令により設定された座標の計算、G02やG03による円弧加工に対応した円弧の中心位置の座標計算など容易ではない。

そこで、NCデータの変更修正、あるいはグラフィック表示された工具軌跡を指示して、NCデータを変更修正できる、両方向からの変更修正が可能なソフトの開発を検討することとし、開発を進めている。これまでの開発において、NCデータの編集機能の一部を開発できたので報告する。

キーワード：CG, NCデータ, CAD/CAM

1 はじめに

三次元CAD/CAMシステムを利用して作成したNCデータの容量は、一般的に500 KB (約25000行程度) を超えるデータとなることが多い。このように約20000行を超えるようなNCデータの修正編集は、修正箇所の探索だけでも容易ではなく、かつNCデータを直接編集しようとする、相対指令により設定された座標の計算、G02やG03による円弧加工に対応した円弧の中心位置の座標計算など容易ではない。

さらに、NCデータを基に工具軌跡がグラフィック表示されている場合には、NCデータの変更修正に伴い工具軌跡も変更して表示されることが望ましい。このような機能を有するNCデータの編集が可能なソフトは見あたらない。

そこで、NCデータの変更修正、あるいはグラフィック表示された工具軌跡を指示して、NCデータを変更修正できるソフトの開発を検討することとし、開発を進めた。この開発過程において、DirectXやOpenGL等の機能の評価を実施したので報告する。

2 開発方法

NCデータリストを表示し、それらのデータの一部を指定すると、そのデータが示す工具軌跡部分を色を変更してグラフィック表示することを実現した。しかし、グラフィック表示部分をマウスで指定して、その指示されたNCデータを認識する機能を開発できていない。

この機能の開発にかなりの時間を費やしているが問題点も多いため、編集機能の開発を先に実施することとした。

ここで、NCデータをArrayListクラスとして定義し、編集においては、このクラスの複製を作成し、その複製データにおいて、作業を行うようにした。このようにすることで、作業途中での中断や、誤作業の対応も可能とした。

2-1 OpenGLによる開発

OpenGLを利用して開発したソフトにおいて、NCデータを読み込んだ後、工具軌跡をグラフィック表示させたものを図1に示す。また、読み込んだNCデータのリストを表示しているFormを図2に示す。

NCデータリストの一部をマウスで指示すると、その指示された部分のグラフィック表示部が青色

* 設計生産工程の高効率化に関する研究 (第3報)

*1 機械電子部

に表示される。

この機能において、データが27000行程度の量になると、座標軸の回転などの操作において、動作が遅滞し、操作性が非常に悪くなった。拡大すると視点位置に形状が近づくようになり、ある程度以上になると視野から消失する。また、その他の動作も反応が極めて遅くマウスの動きに追従できないなどの問題点がある。

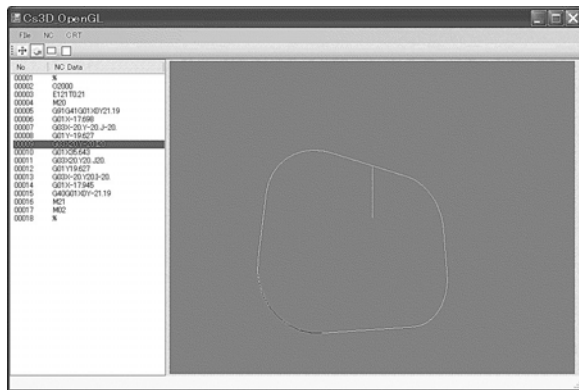


図1 OpenGLによる表示機能

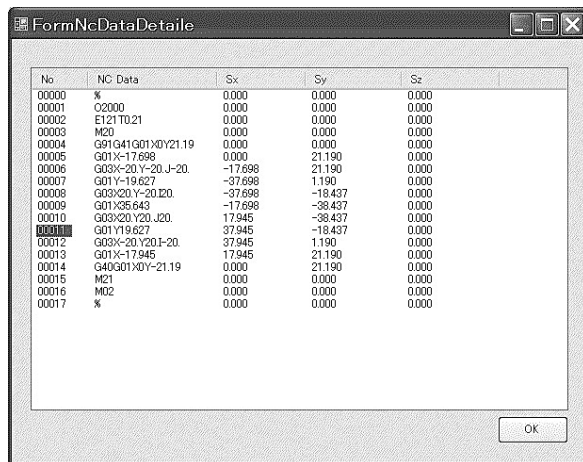


図2 NCデータのリスト表示

開発環境を以前のMicrosoft Visual Studio C#2005からC#2008に更新したところ、マウスで指示されたグラフィック表示部が青色で表示されるべきであるが、適正に動作しなくなり、かつデータ量が增大すると、座標軸などの回転機能において、動作が遅滞するなどの現象が発生し、操作性が劣化した。

2-2 DirectXによる開発

DirectXを利用して開発したソフトにおいて、

NCデータを読み込んだ後、工具軌跡をグラフィック表示させたものを図3に示す。また、読み込んだNCデータのリストを表示しているFormを図4に示す。

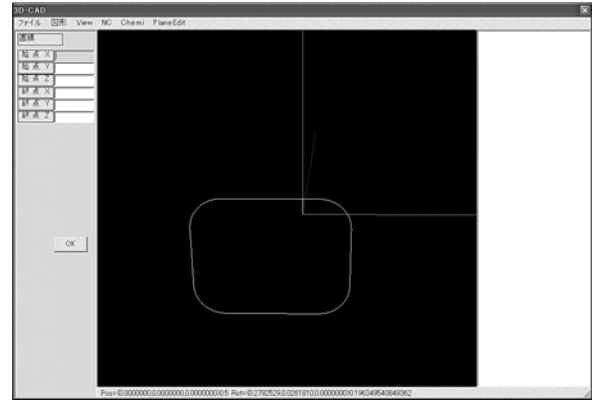


図3 DirectXによる表示機能

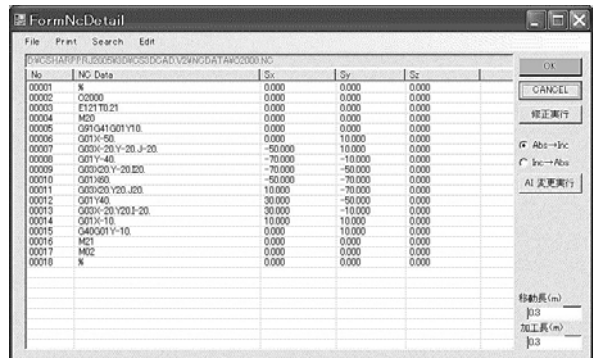


図4 NCデータのリスト表示

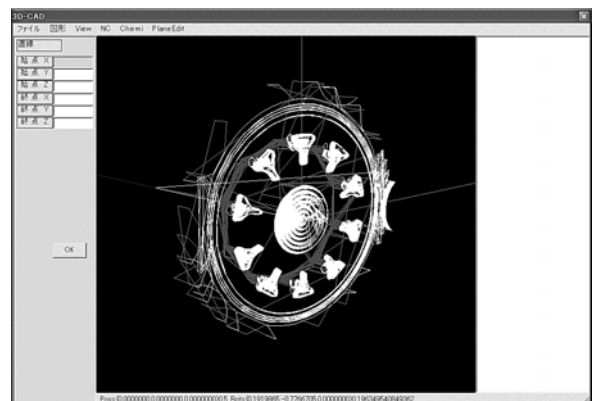


図5 DirectXを利用したプログラムでのNCデータ工具軌跡の表示

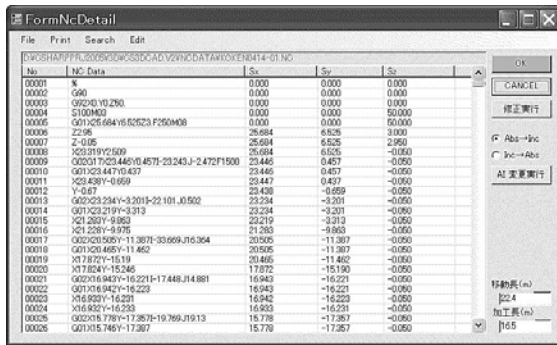
短いデータの場合は、NCデータを読み込んで表示するまでの時間において大差はないが、図5に示すような約27000行程程度のデータ(573KB)の場合は、OpenGLソフトの場合で46秒、DirectXソフトの場合で20秒の、約2倍以上の差を生じた。このような差は、グラフィック表示機能の処理速度の相違により生じているものと推測するが、ソフトにおいて改善の手法が見出せない。

さらに、DirectXを利用して開発したソフトにおいて、図5に示すNCデータを読み込んだ後、

そのデータを絶対座標系(ABS)から相対座標系(INC)へと変換する機能を開発し実施してみたところ、37秒を要した。

その変換前の状態を図6に、変換後の状態を図7に示す。

さらに、図8に示すようなポケット加工のNCデータを図9に示す。このとき、NCデータの一部を選択し、ポップアップメニューで「削除」を指定し(図10参照)、実行した状態のグラフィック表示を図11に示す。



3 結果および考察

プログラム開発はMicrosoft Visual Studio, NET C#を利用した。DirectXによるプログラムでは、デバッグ状態においてもマウスに追従した動作が可能であるが、OpenGLを利用したプログラムでは、マウスに追従した動作が容易ではなく、画面のちらつきも発生した。ただし、実行プログラムを起動した場合には、このような問題は解決される。

今回は、約570KB(約27000行)程度のNCデータを読み込み、NCデータのリストと工具軌跡を表示し、座標軸の回転などの機能をDirectXやOpenGLを利用し、それぞれの動作における操作性について評価できた。

さらに、NCデータの変換機能の一部を開発し、絶対座標系から相対座標系への変更修正、またその逆の変更修正を実施した。その結果、変更修正機能に問題がないことを確認し、データの読み込みと表示において処理速度と操作性の差異を明確にすることができた。

4 まとめ

今回の開発において、座標軸回転などの動作や操作性にやや難点があるものの、NCデータの編集機能の一部などを開発できた。機能の点からはOpenGL利用しているプログラムの方が、コマンドも充実しているために優れているが、動作の観点からは、DirectXを利用したプログラムの方が操作も容易であることが確認できた。

5 参考文献

- 1) 大川善邦他：DirectX9 実践プログラミング，工学社（2004）
- 2) 新藤義昭，阿部正平：OpenGLリアルタイムC#プログラミング，秀和システム（2001）
- 3) 第二I/O編集部編集：I/O BOOKS [書籍版] DirectX9 実践プログラミング，工学社