

部分的な再描画による NC 加工プロセスの高速なアニメーション

乾 正知[†] 高野 誠人^{††}

切削加工におけるトラブル防止には、アニメーション表示により、加工のプロセスをあらかじめ確認しておくことが有効である。NC 切削加工のアニメーションでは、工具がある一定の距離移動するたびに加工シミュレーションを実行し、得られた工作物の形状を画面に表示する処理を繰り返す。工具移動により除去されるのは工作物のごく一部なので、連続的に描画される工作物の画像も、その前後の違いは局所的である。この性質に基づいて、工作物の形状変化に関係する部分だけを選択的に描き直すことで、加工アニメーションを高速化する技術を開発した。描き直し処理には、3次元グラフィックスハードウェアの、隠面消去機能やフレームバッファ間の画像の転写機能を利用する。評価システムを試作し計算実験を行うことで、手法の有効性を検証した。

Fast NC Milling Animation Using Partial Redrawing Technique

MASATOMO INUI[†] and MAKOTO TAKANO^{††}

Animation of the NC milling process is helpful for detecting potential problems of a cutter path prior to actual machining. In the animation process, machined shape of the workpiece is repeatedly computed and displayed after every short motion of the cutter. A single cutter motion usually removes very small portion of the workpiece, therefore most part of two successive pictures in the animation becomes the same. Based on this characteristic of the pictures, the authors develop a fast algorithm for animating the NC milling process. This algorithm efficiently generates a new picture of the workpiece by erasing and redrawing a part of the latest picture corresponding to the removed portion of the workpiece. These operations can be accelerated by using the polygon rendering hardware. An experimental system is implemented and some animations of complex milling processes are demonstrated.

1. はじめに

ダイキャスト製品などの製造に用いられる金型には、複雑な曲面形状を有するものが多い。このような金型のほとんどは、先端に半球形の切刃が取り付けられた、ボールエンドミルを用いて切削加工される。金型加工では深く彫り込むことが多いため、剛性に優れる3軸の数値制御(NC)工作機械の利用が一般的である。以下の議論では、このボールエンドミルによる3軸NC加工のみを対象とするが、提案する技術は容易に他形状の工具を用いた切削加工に適用できる。

1.1 加工プロセスのアニメーション表示

曲面を多用した製品の加工では、ボールエンドミルは膨大な回数の微小な直線移動を繰り返す。曲面加工用のCAMシステムは、加工対象製品の幾何モデルに

基づいて、工具位置の基準となる点(通常は半球形の切刃の中心)の移動経路を計算する。多くのシステムには、工具経路の線画表示機能しか用意されていないため、経路に問題が生じていても、それらを事前に把握することは困難であった。

切削中に発生しうる、工具の工作物への衝突や過大な除去量などの問題を事前にチェックするには、加工の過程をアニメーション表示し、様々な視点から観察することが有効である。加工前の工作物形状を表す立体モデルから、移動するボールエンドミルの掃引形状を差し引いたモデルを計算することで、加工プロセスを幾何的にシミュレーションできる。工具がある一定の距離移動するたびにシミュレーションを実行し、得られた工作物の形状を連続的に描画すれば、加工の過程がアニメーション表示される^{5),8),14),15)}。

すでに多数の加工シミュレーション技術が開発されているが、工作物のZ-map表現に基づく手法が最も実用的と考えられる^{2),6)}。この手法では、Z軸が工具の回転軸方向と一致するように配置された直交座標系を考え、工作物の形状を、xy平面上の均一な直交格

[†] 茨城大学工学部
Faculty of Engineering, Ibaraki University

^{††} 茨城大学理工学研究科
Graduate School of Science and Engineering, Ibaraki University

子に基づいて定義された, z 軸方向に細長い直方体の集積により近似表現する(図 1 参照). 連続した直線移動に対応する, ボールエンドミル先端の切刃の掃引形状は, 球面と円筒形を交互に結合したものになる. 切削加工後の工作物の形状は, これらの球面と円筒形で, 直方体群の上部を切り詰めることで計算できる. Z-map 表現では, モデルの上面に段差が残るため, 滑らかな陰影表示が得られない. 隣接する 4 個の直方体の上端に 2 枚の微小な三角形を貼り付け(図 2 参照), 段差を平面で補間して描画することが一般的である²⁾.

格子間隔 w を小さくするほど, 精密な可視化が可能になる. しかし $1/w$ の平方に比例して直方体数が増加するため, 必要な記憶容量が膨大なものになる. 表示画面の解像度以上に細かい直方体群へ分割しても, 細部がつぶれてしまうため画質は向上しない. そこで本研究では, 格子の総数がウィンドウのピクセル数と同程度になるように, 格子間隔を調整した.

1.2 本論文の目的と構成

加工プロセスのアニメーション表示を実現するためには, 加工の進行にともない変化する工作物の形状を, 毎秒 20~30 回の速度で更新し描画する必要がある. Z-map 表現に基づく加工シミュレーションでは, 工作物の形状は数十万枚のポリゴンで表現される. 最新のグラフィックス表示装置を用いても, これほど多数のポリゴンを描画するにはかなりの時間を要する. そのため素朴な手法では, 画像の切り替わりが遅くなり, 加工の進行を滑らかに表示できない.

工具の各直線移動により除去されるのは, 工作物のごく一部なので, アニメーション表示のために連続的に描画される工作物の画像も, その前後の違いは局所的である. この画像間の類似性に基づき, 工作物の形状変化に関係する部分だけを選択的に描き直すように, 加工アニメーションのアルゴリズムを改良した. 描き直しには, 3 次元グラフィックスハードウェアの, 隠面消去機能や画像の転写機能が利用できる. その結果, NC 切削加工のプロセスを, 高速にアニメーション表示する技術を実現したので報告する.

本技術を用いると, 3 次元グラフィックス処理が強化されたパーソナルコンピュータであれば, 工具の移動回数が十万を超える複雑な曲面加工の全過程を, ごく短時間(数分程度)で表示できる. 工作物の形状変化を表す画像は, 毎秒 30 回程度の速度で更新されるので, 得られるアニメーションは十分に滑らかである. 画面に表示される工作物の形状は, 多数の微細なポリゴンにより精密に表現されているので, 観察者は工作物の状態をリアルに認識できる.

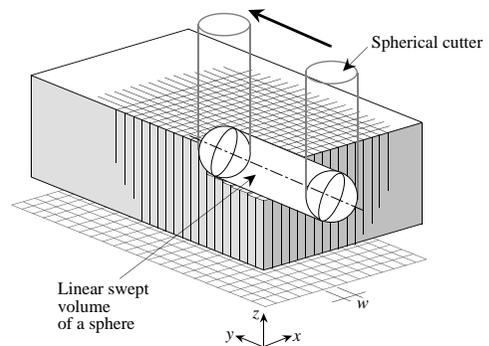


図 1 Z-map 表現に基づく 3 軸 NC 加工のシミュレーション
Fig. 1 Illustration of the Z-map based simulation of 3-axis milling.

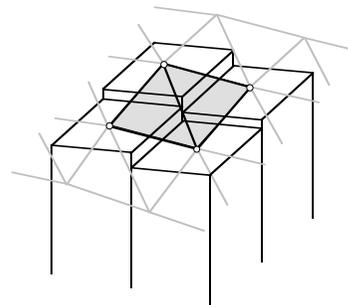


図 2 隣接する 4 本の直方体上端に貼り付けられた 2 枚の三角形面
Fig. 2 Two small triangular polygons attached to the top of four mutually adjacent cubes.

本章では, アニメーション表示の高速化に関する過去の研究について述べる. 3 章では, 切削工具の位置変化の情報に基づいて, 工作物の画像から, その変化しうる範囲を適切に検出し, 描き直す技術について説明する. 自然なアニメーション表示のためには, ダブルバッファの利用が不可欠である. 4 章では, 3 章で提案した技術を, ダブルバッファと組み合わせて用いる方法について議論する. 5 章には, 試作したプログラムによる数値実験の結果を示す.

2. 他研究との関係

滑らかな加工アニメーションを実現するためには, 工作物の形状を高速に表示する技術が重要である. 3 次元コンピュータグラフィックスでは, 立体を多面体化して描画する. 表示速度を改善するには, 画像に悪影響が生じない範囲で, 描画する多角形の数をできるだけ低く抑えればよい. この問題は, 仮想現実感やウォークスルーの分野でよく研究されており, 以下に示すような技術が開発されている¹⁰⁾.

見えない多角形の抽出(Culling) 観察者とは逆方向を向いている多角形の抽出¹¹⁾や, 多面体群の階

層構造を用いて視界の外の多角形を効率的に抽出する技術¹³⁾、他の立体に隠蔽されて観察者からは見えない多面体を抽出する技術¹⁶⁾など。

多重詳細度 (Levels of Detail) 観察者から離れて配置された立体は、少数のピクセルにより表示されるため、細かい形状は識別できない。そこで各立体を、詳細度の異なる複数の多面体で表現しておき、観察者からの距離に応じて表示する多面体を取り替え、描画速度を向上させる^{3),9)}。

イメージへの置き換え 観察者から十分に遠方に配置されていて、その見え方がほとんど変化しない立体については、それらを描いた画像を用意しておき、立体の代わりに画像を表示することで高速な処理を実現する^{1),12)}。

帯状や扇状の多角形群への変換 隣接する三角形や四角形群を、帯状や扇状に接続する形状としてまとめて扱くと、図形間で共有されている頂点の処理が1度で済むため、高速な描画が実現できる⁴⁾。

上述した技術は、建築物や固定された設備などの変形しない立体を、連続的に変化する視点から観察する場合を想定している。一方、切削加工のアニメーションは、観察者の視点はあまり変化しないが、工作物の形状が連続的に変化するという、正反対の性質を持っている。そのため従来技術を用いて、加工アニメーションを高速化することは難しい。

本研究では、切削加工における工作物の形状変化が局所的であることを利用して、形状変化が影響する画像の一部だけを書き直し、残りの部分は以前の画像をそのまま表示することで、アニメーション速度を改善する。部分的な再描画という考え方は、文献14)の古典的な加工アニメーション手法でも採用されているが、この手法では、隠面消去処理や色の計算をピクセルごとに独自の方法を用いて行う必要があり、グラフィックスハードウェアの機能が利用できない。

加工アニメーションに関する他の研究は、描画処理の高速化について、ほとんど議論していない。高速表示を特徴とする商用システムもいくつか知られているが、それらの技術的な内容は明らかにされていない。また実際に表示に要する時間を比較すると(計算機環境や計算条件が異なるので正確な比較とはいえないが)、本論文の提案する技術のほうが、はるかに高速である。

3. 部分的な再描画による加工アニメーション

以下の説明では、切削加工の過程を、斜め上方から見下ろすように観察する場合を考える。3軸NC加工

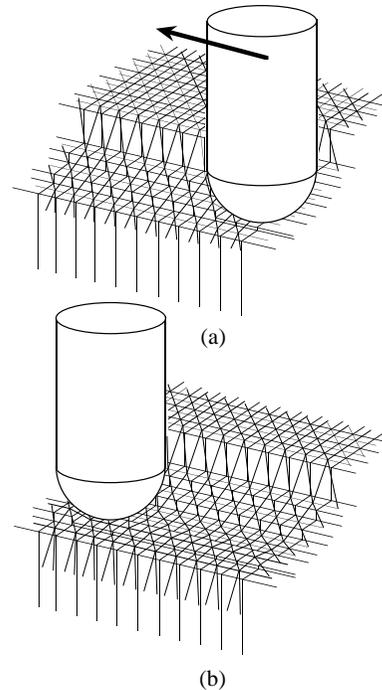


図3 工作物モデルの加工前の状態 (a) と加工後の状態 (b)
Fig. 3 Two states of a workpiece Z-map model. Before a milling simulation (a), and after it (b).

では、工作物は上から削り取られていくので、このように制限しても観察に不都合は生じない。視点は固定されており処理中に変化しないこと、また工作物の形状が平行投影により描画されることを、とりえず仮定する。視点の変更や透視投影の扱いについては、最後の節でまとめて議論する。

3.1 処理の流れ

工具が距離 d 移動するごとに工作物の画像を更新し、アニメーション表示することを考える。具体的には、半径 r のボールエンドミルの基準点が $p_0 = (x_0, y_0, z_0)$ にあるときの工作物の形状が、現在画面に描かれているものとし、この点からの工具の移動距離を道のりに積算する。工具が点 p_1, p_2, \dots, p_{n-1} を経由して点 p_n に到達したとき、移動距離が d を上回ったなら、この点に至るまでの加工プロセスをシミュレーションし、得られた工作物の形状を画面へ描画し直す。

図3(a)には、工作物のZ-mapモデルを構成する直方体群が、工具の移動により切り詰められる前の状態を、また(b)には切り詰められた後の状態を示した。各状態の工作物の形状は、直方体の上に貼り付けられている三角形を描くことで、画面に表示される。

加工後の形状のZ-mapモデルに新たに貼り付けられた三角形群を、加工前の形状の画像に上書きするだ

けでは、希望する結果は得られない。3次元コンピュータグラフィックスでは、他の図形に隠されて観察者から見えない図形（隠面）を、デブバッファ技術を用いて画面から消去する。切削加工は素材を除去するプロセスなので、加工後の形状を表す三角形は、必ず加工前の形状を表す三角形の下側に存在する。このような面は基本的に隠面として扱われるため、これらを描画しても自動的に消去されてしまい、表示結果に反映されない。

したがって、部分的な描き直しにより加工後の形状を表示するためには、以下の2段階で処理を進める必要がある。

Step 1 加工前の形状を表す三角形群のうち、工具移動により切り詰められた直方体に貼り付けられていたものの画像を、画面から消去する。

Step 2 画像の消去によって生じた「穴」を覆うように、加工後の工作物の形状を表す三角形の画像を上書きする。

以下の節では、各ステップでの処理について詳しく述べる。

3.2 加工前形状の画像の消去

Z-map モデルを構成する直方体は微細なため、工具が少し移動するだけで、かなりの本数の直方体が切り詰められる。それらに貼り付けられている三角形の数は膨大なものになるため、その画像を個別に画面から消去するには、大きな手間を要する。本研究では、これらの三角形をすべて包含する x, y, z 軸に平行な姿勢のボックスを用意し、この画像を消去することで、ボックス内の全三角形の画像を画面から1度に除く。

直方体が工具移動により切り詰められるためには、その対応する xy 平面上の格子点が、工具の移動形状の xy 平面への投影に含まれている必要がある。そこでボックス形状を、その xy 平面への投影が、ボールエンドミルの移動形状の投影を包含するように定義する。各三角形面は、隣接する直方体を跨るように貼り付けられているので（図2参照）、ボックスの投影は、切り詰められた直方体群の隣の直方体も覆うように、一回り大きく定義する。

以上の議論から、ボックスの x 軸方向の範囲は、 $[\min(x_0, x_1, \dots, x_n) - r - w, \max(x_0, x_1, \dots, x_n) + r + w]$ 、また y 軸方向の範囲は、 $[\min(y_0, y_1, \dots, y_n) - r - w, \max(y_0, y_1, \dots, y_n) + r + w]$ となる。ただし x_i, y_i は工具の通過点 p_i の x, y 座標値を、また w は Z-map モデルの直方体の太さを意味する（図1参照）。ボックスの z 軸方向の範囲は、上述のように定義された xy 平面上の矩形領域内に、対応する格子点

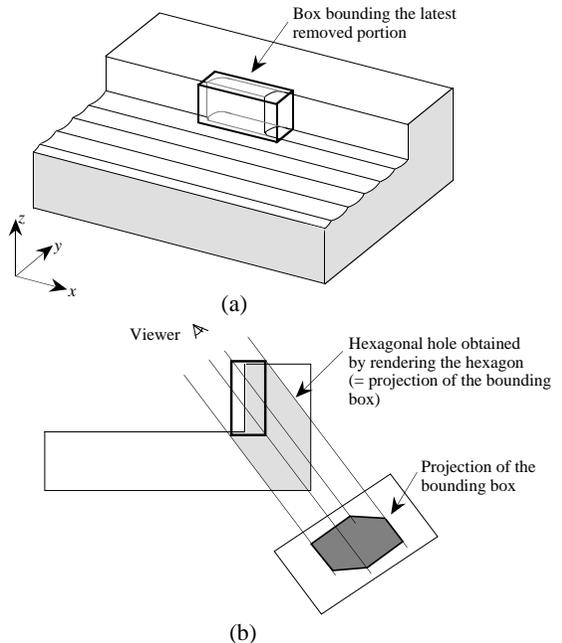


図4 加工により除去された部分の画像を消去するために定義されたボックスの例

Fig. 4 Definition of a bounding box for erasing a part of the picture corresponding to the latest removed portion of the workpiece.

を持つ直方体群の切り詰め前の高さを調べ、それらの最小値と最大値を含むように決めればよい。

図4(a)に、このようにして得られたボックスと、工作物の形状を重ね合わせた様子を示した。工作物の画像からボックス部分を消去する処理には、隠面消去技術が利用できる。多面体の隠面消去は、通常とは逆に、観察者により近い図形を消去し、より遠い図形を表示する目的にも使える¹¹⁾。そこで図4(b)に示すように、十分遠方に視線と直交する向きの平面を用意し、そこにボックス形状を平行投影した六角形を定義する。そして、より遠い図形を表示するように一時的に隠面消去の指定を変更したうえで、六角形を画面の背景色で描く。するとその手前の図形がすべて消去され、工作物に、ボックスの投影に対応する、六角形の穴が開いた画像が表示される。

3.3 加工後の形状の再描画

次に六角形の「穴」を塞ぐように、加工後の形状を上書きする。すなわち、加工後の形状を表す Z-map モデルに貼り付けられたすべての三角形について、六角形の穴との位置関係を調べ、穴を視線から一部でも覆い隠すように三角形が配置されている場合には、その画像を描くという処理を繰り返す。頂点が1つでも六角穴の内部にある三角形や、辺が1本でも六角穴と

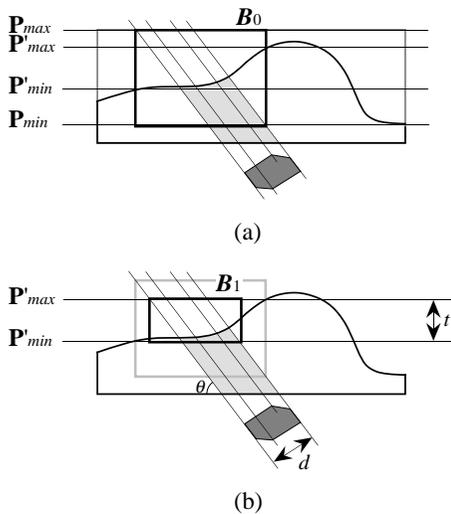


図5 ボックス B_0 とボックス B_1 の定義

Fig. 5 Definitions of two bounding boxes B_0 and B_1 .

交差している三角形を選び、その画像を描画することになる。工作物形状を表す膨大な数の三角形について、この被覆条件の判定と描画処理を単純に繰り返したのでは、高速な表示は望めない。

本研究では、以下に示す2段階のボックスチェックを用いて、再描画が必要な三角形を事前に絞り込むことで、処理を効率化する。準備として、加工アニメーションの前に、初期状態の工作物モデル上面の最高点を調べ、その点を通る水平な面 P_{max} を計算しておく。また最低点の高さを調べ、その値を変数 h_{low} に保存しておく。以後、加工シミュレーションにおいてボールエンドミルが移動するたびに、切削の下端の高さを調べ、その値が h_{low} よりも小さいときには、 h_{low} をその値へ更新する。

Step 2.1 xy 平面に平行で、高さ h_{low} の平面 P_{min} を計算する。 h_{low} の値は、工作物上面の高さの下限を表している。したがって、工作物の上面を覆うすべての三角形は、2枚の平面 P_{max} と P_{min} の間に存在する。 P_{min} と P_{max} の定義する平行な板形状と、六角穴の補集合に相当する六角柱の積形状を計算し、それを包含する x, y, z 軸に平行な姿勢のボックス B_0 を得る(図5(a)参照)。再描画が必要な三角形は、ボックス B_0 に必ず含まれる。そのような三角形は、 B_0 の xy 平面への投影を用いることで、容易に選択できる。

Step 2.2 ボックス B_0 が包含する三角形群の全頂点を調べ、それらの中で z 座標値が最大の点と最小の点を得る。 z 座標値が最大の点を通る xy 平面に平行な平面として P'_{max} を、また z 座標値

が最小の点を通る平面として P'_{min} を定義する。 P'_{max} は P_{max} と一致するかその下側に存在し、 P'_{min} は P_{min} と一致するかその上側に存在する。再描画すべき三角形は、 P'_{max} と P'_{min} の間に存在する。そこで P'_{max} と P'_{min} が定義するより薄い板形状と六角柱の積を計算し、それを包含するボックス B_1 を定義する(図5(b)参照)。

金型の仕上げ加工のような、深く掘り込まれた形状の加工では、ボックス B_1 はボックス B_0 よりもかなり薄く小さなものになる。再描画する三角形面の数が少なければ少ないほど、加工後の工作物の形状を速く表示できる。本研究では、ボックス B_1 を用いて三角形面を少数に絞り込み、それらの中で六角穴の被覆条件を満たすものを選択し描画することで、高速なアニメーション表示を実現した。

三角形面の絞り込みに用いられるボックスのサイズは、六角形の穴の傾き、すなわち工作物形状を見下ろす角度に大きく依存する。穴の最大径を d 、 P'_{max} と P'_{min} のなす板形状の厚みを t 、穴と水平面となす角度を θ とすると、生成されるボックスの幅 w は、

$$w = \frac{d}{\sin \theta} + \frac{t}{\tan \theta} \quad (1)$$

まで広がりうる。したがって本手法では、角度 θ が0に近づくと、ボックスが非常に大きくなるのが分かる。 θ が0に近づくにつれて、六角穴と交わる三角形数も急速に増加するので、結果として高速なアニメーション表示は困難になる。

3.4 手法の改良

提案した手法を改良し、視点変更や透視投影による描画などを実現する方法について簡単に述べる。

視点の変更 本手法は視線方向に依存しているため、アニメーション中の視点変更には対応できない。ただしアニメーションを中断し、視点を変更したうえで、加工途中の工作物の形状を画面に表示し直し、アニメーションを再開することは可能である。この一連の処理において最も手間を要するのは、加工途中の工作物形状の表示だが、グラフィックスハードウェアを利用した高速な手法が開発されており⁷⁾、待ち時間はほとんど問題とならない。

透視投影 提案した手法は、若干の修正を施すだけで、透視投影による描画に対応できる。透視投影の場合、画像の消去時に生じる「穴」の形状が、視点を頂とする六角錐になるので、ボックス B_0 や B_1 を定義する際には、六角錐と板形状の積を計算することになる。2つの多面体の積を計算する手間は、面と面の交線計算の回数に比例する。六角錐

と板形状の積を計算する際の際の交線計算の回数は、六角柱と板形状の積を計算する場合と同じになる。得られる積形状や、その形状を包含するボックスのサイズもほぼ同じになるので、投影法を変更してもアニメーション時間はほとんど変化しない。

工具の移動距離の管理 同じ距離移動しても、半径の大きい工具ほど切り詰める直方体の数が増えるため、再描画を要する三角形面の数が増加し、表示速度が低下する。工具半径によらず一定の表示速度を維持するために、画像更新の基準となる工具の移動距離に補正を加えることにした。工具の直線移動により切り詰められる直方体の本数は、工具半径 r にほぼ比例する。そこで与えられた距離 d ではなく、それを工具半径で除した値 d/r を基準に画像の更新を行うことで、表示速度をできるだけ一定に保つ。

4. ダブルバッファの利用

前述した技術をそのまま利用して処理を行うと、部分形状の消去や描き直しの過程が画面に表示されるため、不自然なアニメーションになってしまう。3次元コンピュータグラフィックスでは、このような問題の生じない滑らかなアニメーションを得るために、ダブルバッファを用いる。この手法では2枚のフレームバッファを用意し、それぞれを「表」と「裏」に置く。画面には、表に置かれたバッファの内容のみが表示され、裏に置かれたバッファへ図形を描く過程は画面に現れない。そこで図形を裏側のバッファに描き終えてから、それを表側のバッファと交換し表示することで、画像のスムーズな切替えを実現する。

4.1 ダブルバッファ利用における問題点

部分的な再描画技術を用いた加工アニメーションは、そのままではダブルバッファに対応できない。この問題を、ボールエンドミルが経路上の点 q_0, q_1, \dots, q_n を通過するごとに、工作物形状を表示し直す場合を例に説明する。2枚のフレームバッファを、それぞれ A-バッファ、B-バッファと名付ける。最初は A が表側、B が裏側のバッファとする。

工具が q_0 から q_1 に至るまでの工作物の形状変化は、まず B-バッファに描かれる。次にバッファが交換され、今度は A-バッファに、工具が q_1 から q_2 に至るまでの工作物の形状変化が描かれる。

この処理を繰り返すと、A-バッファには、

$$q_1 \rightarrow q_2, q_3 \rightarrow q_4, q_5 \rightarrow q_6, \dots$$

の工具移動に対応する画像が描かれ、一方 B-バッファには、

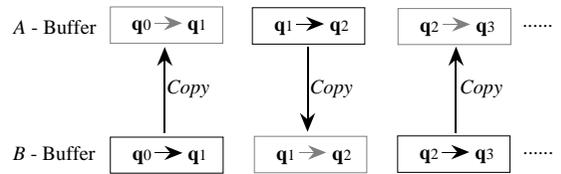


図6 各バッファに新たに描かれた部分を他方のバッファへ転写し、両方のバッファの内容を同一に保つ操作

Fig. 6 Copying a part of the workpiece picture drawn in a buffer to another buffer so that the pictures appearing in two buffers become the same.

$$q_0 \rightarrow q_1, q_2 \rightarrow q_3, q_4 \rightarrow q_5, \dots$$

という移動に対応する画像が描かれるため、各バッファには交互に描かれない部分が生じてしまう。

4.2 画像の転写による解決

グラフィックスハードウェアには、片方のフレームバッファに描かれた画像上の矩形領域を、他方のフレームバッファの対応する位置へコピーする機能が用意されている。この機能を用いて、片方のバッファに描かれていて、他方には描かれていない工作物の画像を互いに転写し合うことで、前述の問題を解決する。

最初のバッファ交換後に、表側の B-バッファに描かれている、工具移動 $q_0 \rightarrow q_1$ に対応する領域の画像を、裏側の A-バッファへコピーする。そのうえで、A-バッファに工具移動 $q_1 \rightarrow q_2$ による形状変化を描く(図6参照)。バッファの交換後、今度は A-バッファに描かれている工具移動 $q_1 \rightarrow q_2$ に対応する画像を、B-バッファへコピーする。以上の処理を交互に繰り返すと、どちらのバッファにもつねに工作物の完全な形状が描かれているようになる。

画像をコピーする範囲は、各工具移動について、加工後の形状を再描画した領域で十分である。そこでバッファの交換後に、直前の再描画時に定義したボックス B_1 を画面へ投影した図形を考え、それを包含する矩形領域を表側のバッファから裏側のバッファへコピーする。画像のコピーに要する時間は、コピーする矩形領域の広さにほぼ比例する。工具の移動量が小さい場合には、 B_1 を投影した図形の面積もわずかとなるので、このコピー処理による描画速度の低下は無視できるほど小さい。

5. 計算実験

提案した手法に基づいて、切削加工のプロセスをアニメーション表示するプログラムを試作し、計算実験を行った。実験には、直方体の総数が 728×728 本になるように格子間隔を調整した Z-map モデルを用いた。工作物の形状を表す三角形面の総数は約 100 万枚とな

表 1 アニメーション速度
Table 1 Animation speed.

| Number of blocks | Tool radius (mm) | Animation time (sec.) | Average frame rate (frames/sec.) |
|------------------|------------------|-----------------------|----------------------------------|
| 9542 | 5.0 | 94.96 | 23.69 |
| 35391 | 3.0 | 228.63 | 33.39 |
| 126719 | 1.0 | 163.91 | 38.72 |
| 156799 | 1.0 | 272.27 | 44.47 |

る。プログラムの実行には、PentiumIII (450 MHz) を CPU とする、主記憶 476 MB のパーソナルコンピュータを用いた。このコンピュータには、Cobalt とよばれる 3 次元グラフィックス表示用のハードウェアが組み込まれている。

金型加工用に生成された 4 種類の工具経路データを用いて、プログラムの性能評価を試みた。実験結果を表 1 に示す。これらの実験では、前述の議論に基づき、画像更新の基準となる工具の移動距離を、その値と工具半径の積が 10.0 mm となるように調整した。また、工作物を観察する視点を水平面から 60 度斜め上方に設定し、平行投影により画像を描画した。表の最初の列と 2 列目は、各経路における工具移動の回数（ブロック数）と、ボールエンドミルの半径（単位は mm）を示す。3 列目には、すべての加工プロセスをアニメーション表示するのに要した時間（単位は秒）を示す。また 4 列目の数値は、1 秒間に描画された画像数（フレームレート）の平均を表す。図 7 に、最初の行に対応する工具経路と加工後の工作物の形状を、設定した視点から観察した画像を示す。

以上のデータから、提案した手法により 10 万ブロックを超える複雑な加工の全プロセスを数分で表示できること、また平均フレームレートはいずれの場合も 20 を上回っており、滑らかなアニメーション表示が得られていることが分かる。毎回工作物を構成するすべての三角形を描き直した場合には、平均フレームレートは 1 前後であることから、本手法を用いることで切削加工アニメーションを、20 倍以上高速化できることが分かる。

加工プロセスを観察する際の、視線と水平面のなす角度（図 5 中の角度 θ ）を 90 度から 10 度刻みに減らしたときの、処理時間の変化などを表 2 に示す。この実験では、表 1 の 4 行目の工具経路を用いた。2 列目にはアニメーション時間の変化、3 列目には画像の再描画の際に新たに描かれた三角形数の平均、そして 4 列目には平均フレームレートの変化を示した。この実験から、角度が減少すると描画する三角形数が急激に増加し、アニメーション速度が低下すること、そして

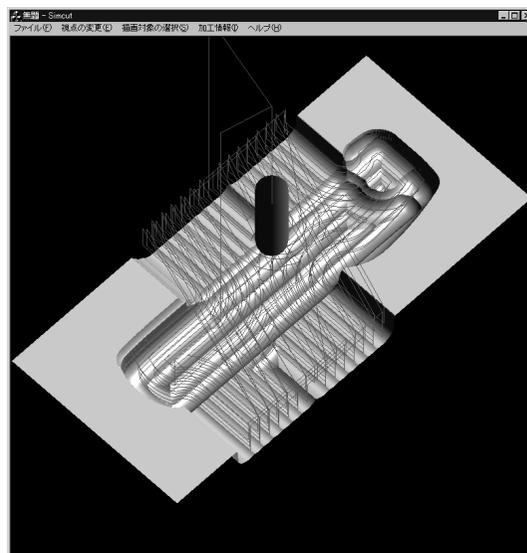


図 7 例題に用いた工具経路と加工後の工作物形状

Fig. 7 An example cutter path and workpiece shape after a milling simulation.

表 2 工作物を見下ろす角度によるアニメーション速度の変化
(156799 ブロックの工具経路の場合)

Table 2 Animation speed change according to the viewing angle (using NC code with 156799 blocks).

| Viewing angle θ (degree) | Animation time (sec.) | Average number of rendered triangles per frame | Average frame rate (frames/sec.) |
|---------------------------------|-----------------------|--|----------------------------------|
| 90 | 202.06 | 3483 | 59.76 |
| 80 | 208.70 | 4523 | 58.01 |
| 70 | 228.82 | 6192 | 52.91 |
| 60 | 263.70 | 8913 | 45.91 |
| 50 | 351.95 | 13825 | 34.40 |
| 40 | 501.52 | 22799 | 24.14 |
| 30 | 806.36 | 41678 | 15.01 |

角度が 30 度よりも小さくなると、滑らかなアニメーション表示が困難になることが分かる。

6. ま と め

本論文では、複雑な NC 加工のプロセスを、高速にアニメーション表示する技術について議論した。この処理では、工具がある一定の距離移動するたびに加工シミュレーションを実行し、得られた工作物の形状を描画する。工具移動により除去されるのは工作物のごく一部なので、連続的に描画される工作物の画像も、その前後の違いは局所的である。この性質に基づいて、工作物の形状変化に係る画像の一部だけを選択的に描き直すことで、高速な加工アニメーションを実現した。描き直し処理には、グラフィックスハードウェアの、隠面消去機能や画像の転写機能を利用する。

提案した技術に基づく加工アニメーションにおいて、加工シミュレーションに要する時間、画像の一部を消去するのに要する時間、再描画範囲の絞り込みに要する時間、三角形の描画に要する時間などを比較したところ、依然として全体の約 9 割の時間を、描画処理に費やしていることが判明した。加工アニメーションをさらに高速化するためには、描画処理の改善が最も効果的なことが分かる。

謝辞 工具経路データの入手に際しては、(株)リコーの田村秀貴氏と(株)リアルファクトリーの原雄司氏にご協力いただいた。本研究の一部は、科学研究費補助金基盤研究(B)(課題番号 11450052)を受けて行われた。この場を借りて深く感謝する。

参 考 文 献

- 1) Aliaga, D.G. and Lastra, A.: Automatic Image Placement to Provide A Guaranteed Frame Rate, *Computer Graphics Proceedings, SIGGRAPH 99*, pp.307-316 (1999).
- 2) Choi, B.K. and Jerard, R.B.: *Sculptured Surface Machining, Theory and Applications*, pp.255-258, Kluwer Academic Publishers, Dordrecht (1998).
- 3) Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P., Brooks, F. and Wright, W.: Simplification Envelopes, *Computer Graphics Proceedings, SIGGRAPH 96*, pp.119-128 (1996).
- 4) Evans, F., Skiena, S. and Varshney, A.: Optimizing Triangle Strips for Fast Rendering, *Proc. IEEE Visualization*, pp.319-326 (1996).
- 5) Huang, Y. and Oliver, J.H.: NC Milling Error Assessment and Tool Path Correction, *Computer Graphics Proceedings, SIGGRAPH 94*, pp.287-294 (1994).
- 6) 乾 正知: 3 軸数値制御工作機械による曲面加工の高速なシミュレーション, *情報処理学会論文誌*, Vol.40, No.4, pp.1808-1815 (1999).
- 7) Inui, M. and Kakio, R.: Fast Visualization of NC Milling Result Using Graphics Acceleration Hardware, *Proc. IEEE Int. Conf. Robotics and Automation*, pp.3089-3094 (2000).
- 8) Kawashima, Y., Itoh, K., Ishida, T., Nonaka, S. and Ejiri, K.: A Flexible Quantitative Method for NC Machining Verification Using a Space-Division based Solid Model, *The Visual Computer*, Vol.7, pp.149-157 (1991).
- 9) Luebke, D. and Erikson, C.: View-Dependent Simplification Of Arbitrary Polygonal Environments, *Computer Graphics Proceedings, SIGGRAPH 97*, pp.199-208 (1997).
- 10) Möller, T. and Haines, E.: *Real-Time Rendering*, A K Peters, Natick, Massachusetts (1999).
- 11) OpenGL Architecture Review Board, *OpenGL Reference Manual*, Second Edition, Addison Wesley Developers Press, Reading, Massachusetts (1996).
- 12) Shade, J., Lischinski, D., Salesin, D., DeRose, T. and Snyder, J.: Hierarchical Image Caching for Accelerated Walkthroughs of Complex Environments, *Computer Graphics Proceedings, SIGGRAPH 96*, pp.75-82 (1996).
- 13) Teller, S.J. and Sequin, C.H.: Visibility Pre-processing For Interactive Walkthroughs, *Computer Graphics, Proc. SIGGRAPH 91*, Vol.25, No.4, pp.61-69 (1991).
- 14) Van Hook, T.: Real-Time Shaded NC Milling Display, *Computer Graphics, Proc. SIGGRAPH 86*, Vol.20, No.4, pp.15-20 (1986).
- 15) Wang, W.P. and Wang, K.K.: Geometric Modeling for Swept Volume of Moving Solids, *IEEE Computer Graphics and Appl.*, Vol.6, No.12, pp.8-17 (1986).
- 16) Zhang, H., Manocha, D., Hudson, T. and Hoff III, K.E.: Visibility Culling using Hierarchical Occlusion Maps, *Computer Graphics Proceedings, SIGGRAPH 97*, pp.77-88 (1997).

(平成 12 年 6 月 21 日受付)

(平成 13 年 1 月 11 日採録)



乾 正知(正会員)

昭和 36 年生。昭和 61 年東京大学大学院工学系研究科情報工学専攻修士課程修了。平成 5 年より茨城大学工学部システム工学科助教授。工学博士。形状処理技術とその機械生産自動化への応用に興味を持つ。平成 4 年井上研究奨励賞受賞。精密工学会, IEEE, ACM 各会員。



高野 誠人

昭和 52 年生。平成 8 年茨城大学工学部システム工学科卒業。現在、同大学大学院理工学研究科システム工学専攻修士課程在学中。複雑な幾何処理を、グラフィックスハードウェアを用いて高速化する技術に興味を持っている。