

CSG モデル向き濃淡面画作成専用ハードウェア・ プロセッサの開発研究†

—システム構成法の集合・写像による定式化—

三上貞芳^{††} 嘉数侑昇^{††}

本論文は、機械部品設計 CAD システムに重要な役割を果たす CSG ソリッド形状モデルの濃淡面画作成処理に関して、その専用ハードウェア化、並列処理の実現のための理論を体系的に展開したものである。具体的には、問題を集合・写像の枠組により、VLSI レベルの機能モジュールの接続による構成の CSG 濃淡面画作成専用ハードウェアの構築問題としてとらえ、それを定式化する以下のような方法を提案している。まず面画作成処理を集合・写像の枠組により表現し、機能モジュールを写像で記述することで、設計を写像の合成問題として定式化できることを示す。次に定式化された合成問題に対し、含まれる写像を任意個の写像へ分解して合成写像へ再表現する操作で、並列処理構成が導入可能になることを示す。この場合必要となる各写像の処理内容・負荷分散・プロセッサ結合方法は、式からこれらを展開することができることを示す。展開した理論を例えば光線探索法に適用することで、問題に対して並列構成を含めた非同期パイプライン処理型の専用ハードウェアが得られることを示している。

1. はじめに

3次元ソリッド形状を計算機内に情報として表現する方法の一つに CSG 表現 (Constructive Solid Geometry) があげられる¹⁾。この表現方法は3次元ソリッド形状の表現として多くの利点や欠点が存在するが、機械部品設計を目的とした CAD システムの形状モデルとして近年その積極的な利用が期待されているものの一つである。ここではその利用に関して最も問題となっている、CAD システムにおけるマンマシンインタフェースの主たる役割を担う濃淡面画作成処理 (以下 CSG-CG 処理と呼ぶ) の高速化問題を取りあげる。

CSG-CG 処理の高速化問題に関して従来の多くの研究から、現状の汎用計算機では処理能力の面で望めず、解決方法として CSG-CG 処理の並列処理化を含めた専用ハードウェア (以下 CSG-CG 専用機と呼ぶ) を構築し、その利用を図ったほうが有効であろうことが指摘されている²⁾。そこで本論では CSG-CG 専用機の開発による問題解決のアプローチをとることを前提とする。

一般に問題向き専用ハードウェアの実現には、その

対象とする問題の処理により利用するハードウェアの要素技術とその構成にさまざまな段階を設定することができる。例えば固定的な処理に対してレジスタトランスフェレレベルでの実現、プログラムの自由度を必要とする処理に対してプロセッサネットワークによる実現等が一般的に考えられている³⁾。

ところで上述の高速化問題とともに CSG-CG 処理の場合、基本的に汎用機上の CAD システムで扱う形状をすべて表現できることが要求されるゆえ、その専用ハードウェアによる実現には汎用機並のプログラム能力が一部必要となろう。これらを解決する一つの方法として、①DSP (デジタル信号処理用プロセッサ) に代表されるような、プログラムが可能であり、かつ特定の処理に向けたアーキテクチャをもつ VLSI (以下機能モジュールと呼ぶ) のレベルをハードウェアの要素として利用し、②それらの並列処理システムを実現する方法による問題解決が考えられる。

換言すれば、このような問題解決法を指向した CSG-CG 専用機の設計は、①機能モジュールに割り当てる処理内容、②機能モジュール間の結合方法、の決定からなる。ところで機能モジュールとなるプロセッサは、ハードウェア技術の進歩に伴い、その性能・用途などの新しいものが次々と作られつつある。一方 CAD 技術の進歩に伴い、処理の量・質・高速性ともにますます要求が増えている CSG-CG 処理には、新しいハードウェア技術、すなわち新しい機能モジュールの早急な導入が必要とされているのが現状である。

† Development of Shaded Picture Oriented Hardware Processor for CSG Models—Formalizing the System Configuration by Set and Mapping Theories—by SADAYOSHI MIKAMI and YUKINORI KAKAZU (Department of Precision Engineering, Faculty of Engineering, Hokkaido University).

†† 北海道大学工学部精密工学科

これらのことを考慮すると、現時点の技術で実現できる具体的な一つの CSG-CG 専用機を作ることのみならず、さまざまな機能モジュールに対してそれを用いた CSG-CG 専用機を作成するための方法論を確立することが、当面必要となる課題となろう。

このような視点のもとに、本論文では、機能モジュール、CSG-CG 処理を形式的に記述することにより、設計問題を容易な形式的操作に帰着させるような方法を模索する。具体的には、集合・写像の枠組を利用して、設計を写像の合成問題として定式化する方法論を提案する。

すなわち、この方法論の目的は、① CSG-CG 処理、機能モジュールの記述に適した記述の枠組を構築し、設計のための道具立てを示すこと、②構築した記述法を用いて CSG-CG 専用機の記述を求める形式的な方法を明らかにすることである。以下 2 章で方法論を展開し、3 章で例として、光線探索法に対して提案した枠組を適用して、非同期パイプライン処理型の専用ハードウェアを設計解として得る過程を示す。

2. 方法論

2.1 諸概念の定義

CSG-CG 専用機の設計に集合・写像の表現による形式的な方法を導入する前提として、CSG-CG 処理が対象とする以下の情報を集合の枠組で記述する。

【定義 1】 (ソリッド形状)

式(1)により 3次元ソリッド形状を定義する。すなわち

$$S \subset \mathbf{E}^3, \quad S^s \subset S, \quad (1)$$

ここで \mathbf{E}^3 は 3次元ユークリッド空間を表し、 S^s の記述は S の境界を表す。また、 S の全体を *SOLID* と表す

【定義 2】 (プリミティブ形状)

パターン関数と呼ばれる実数値関数 F_i ,

$$F_i: \mathbf{E}^3 \rightarrow \mathbf{R}, \quad (2)$$

により、一般にプリミティブ形状と呼ばれる基本形状 P_i を式(3)で定義する、

$$P_i \equiv \{x; F_i(x) \leq 0\} \in \text{SOLID}, \quad (3)$$

ここで \mathbf{R} は実数の全体集合を表す。 F_i は手続き的なものを含む広い意味での実数値関数だが、定義の容易さ、形状処理の容易さなどの理由で通常 2 次関数等の解析的に簡単な式を用いる

例として図 1 に示したプリミティブ形状 P_3 は、パターン関数 F_3 が 0 および負の値をとる空間点の集合

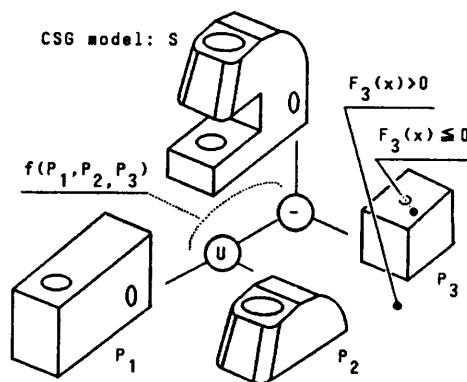


図 1 CSG 表現によるソリッド形状の定義例
Fig. 1 An example of the solid defined by a CSG description.

として定義される。

【定義 3】 (CSG 定義)

形状集合演算式と呼ばれる関数 f ,

$$f: (\text{SOLID})^{N_p} \rightarrow \text{SOLID}, \quad (4)$$

および、 N_p 個のプリミティブ形状 $P_1 \sim P_{N_p}$ を用いて、式(5)により複雑なソリッド形状 S を定義できる。

$$S \equiv f(P_1, \dots, P_{N_p}). \quad (5)$$

f および各 P_i のパターン関数 F_i の組、 $\langle f, F_1, \dots, F_{N_p} \rangle$ をもって、形状 S の CSG 定義とする。また f の全体を *SETP* と定義する

先に示した図 1 の CSG 定義形状の例では、関数 F_1, F_2, F_3 により P_1, P_2, P_3 の三つのプリミティブ形状を与える \mathbf{E}^3 上の集合が定まり、形状集合演算式 $f(P_1, P_2, P_3) = P_1 \cup P_2 \setminus P_3$ により得られる集合が、目的とする CSG 定義形状として定まる。

【定義 4】 (投影面)

ラスタグラフィックス出力装置の画面(投影面)を、離散的な画素の列として式(6)の記述で定義する、

$$\text{SCREEN} \equiv (\text{PIX}_i; i \in \{1, \dots, N_s\}, \text{PIX}_i \in \text{PIXELS}), \quad (6)$$

ここで PIX_i は i 番目の画素、 PIXELS は画素の全体集合、 N_s は画素の個数を表す。さらに、画素の空間的な位置 $\text{pos}_i \in \mathbf{E}^3$ を画素の添え字 i から陽に求める手続き、すなわち、

$$\text{pos}_i = \nu(i), \quad i \in \{1, \dots, N_s\}, \quad (7)$$

なる関数 ν を陽に表現できることを必要条件とする

【定義 5】 (濃淡面画)

画素と色情報の対の集合として、濃淡面画を式(8)

で定義する.

PICTURE

$$\begin{aligned} &\equiv \{(COL(PIX_i), PIX_i); i \in \{1, \dots, N_s\}, \\ &\quad COL(PIX_i) \in COLOR\} \\ &= (COL(PIX_i); i \in \{1, \dots, N_s\}) \quad (8) \end{aligned}$$

ただし, *COLOR* は対象とするラスタグラフィックス出力装置の色指定方法により表示可能な色全体の集合

2.2 CSG-CG 処理の記述

【定義6】 (CSG-CG 処理の記述)

CSG 定義によるソリッド形状の濃淡面画作成処理, すなわち CSG-CG 処理を, 投影面と CSG 定義を定義域とし, 濃淡面画を値域とする式(9)の写像 *DRAW* として定義する. すなわち,

$$DRAW: SCREEN \times SOLID \rightarrow PICTURE. \quad (9)$$

さらに CSG 定義形状である *SOLID* を, 定義3により形状集合演算式とパターン関数に分解して CSG-CG 処理の記述とする.

$$DRAW: SCREEN \times SOLID^N \times SETP \rightarrow PICTURE \quad (10)$$

2.3 機能モジュールの記述

簡潔なハードウェアにより大量の並列化を期待する方法を目的として, 処理全体で条件分岐, データの帰還がないような, パイプライン処理型の流れとなるような CSG-CG 専用機的设计を目標とする. この場合処理の単位となる機能モジュールはパイプライン処理が適している. このとき処理の記述は, 時間に関する順序列として入出力を記述する簡潔な方法で次のように行うことができる.

【定義7】 (機能モジュールの記述)

機能モジュール ξ が処理可能なすべての入出力を, それぞれ自然数 N の部分集合で順序を導入した, a, b なる時系列で記述する.

$$a = (a_i; i \in (I \subset N)), b = (b_j; j \in (J \subset N)). \quad (11)$$

このとき ξ が処理可能なすべての a, b の全体をそれぞれ A, B と記述して, ξ の動作を写像 $\psi: A \rightarrow B$ により記述する.

また1章の前提より機能モジュールはプログラム可能なものとするから, ξ より扱う入出力が限定された, $A' \subset A, B' \subset B$ なる A', B' に対応する写像 $\psi': A' \rightarrow B'$ も, 写像 ψ と同じ機能モジュール ξ により実現できるものとする

ただし, 実際に写像 ψ を定義する場合, 定義域・値

域の列を具体的に要素を枚挙して定義するのは現実的ではない. 現在のところ, 写像を定義する時点で利用者が定義域・値域を想定することに任せる.

2.4 設計の方法論

以上の諸概念の集合・写像による定式化により, CSG-CG 専用機的设计問題は, 次の四つの部分問題として記述できる.

【問題1】 CSG-CG 専用機に用いる機能モジュールを決定し, そのハードウェアを定め, 動作を写像で記述する問題.

いま, 決定した機能モジュールの写像の全体を H と記述する. 入出力列 a, b で定義される, 機能モジュールの一つ $h: a \rightarrow b, h \in H$ に対し, $h^*: a^* \rightarrow b^*, a^* \subset a, b^* \subset b$ なる写像 h^* を考える. 定義7より h^* は h の機能モジュールで実現できる. そこで h^* の全体を H^* と記述するものとする.

【問題2】 $h^* \in H^*$ を用いて, 写像 *DRAW* に等しい合成写像を作る問題.

与えられた h^* に対し, その定義域・値域を制限した写像も, 同様な議論で h の機能モジュールで実現される. したがって, 写像 h^* の値域を部分集合へ分解する操作で, 並列処理構成を導入できる. すなわち,

【問題3】 求めた合成写像 *DRAW* を, それを構成する各写像 h^* の値域を制限した写像 h' を複数個用いた式に変形する問題.

これにより定まる各写像 h' の定義域は, 並列構成の負荷の配分に相当する.

写像の表現で得られたプロセッサ構成を具体的なハードウェアに対応させることで, 設計は終了する. すなわち,

【問題4】 問題3で得られた式を実現するように, 機能モジュールを接続するハードウェア機構を作成する問題.

以上の枠組に従って具体的な解を求めることは使用者に任せるものとする. また設計解の正当性の検証系に関しては, 多くの問題を含むゆえ, あらためて検討したい.

3. 具体例

2章で示した設計の具体性を検証することを目的として, 簡単なパイプライン構成の CSG-CG 専用機を設計する過程を以下で討論する.

3.1 問題1: 機能モジュールの決定と記述

CSG-CG 処理を記述する写像 *DRAW* の定義域・

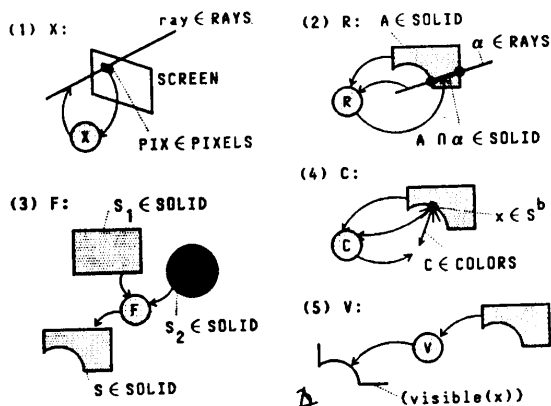


図2 機能モジュールの処理内容
Fig. 2 Processes of the functional modules.

値域は、4種類の集合、*SCREEN*, *SOLID*, *SETP*, *PICTURE* からなる。個々の機能モジュールを特定の種類の処理に専用化する目的では、それぞれが処理対象とする集合の種類はなるべく少ないことが望ましい。そこで以上の四つの集合をそれぞれ処理対象とする機能モジュールを、次のように決定する。(以下の機能モジュールの処理内容の説明は、図2を参照のこと。)

(1) 集合 *SCREEN* (投影面) を対象とする機能モジュール: 写像 *X*

①処理: *SCREEN* を定義域とし、*SCREEN* を構成する要素 $PIX \in PIXELS$ に対して、*PIX* と視点を通る3次元空間直線 $ray \in RAYS$ を与える写像 $X: SCREEN \rightarrow RAYS$ で定義する。ここで *RAYS* は3次元空間直線である視線の全体を表す。

②ハードウェア: 同一の式の繰り返し計算に適した、数値計算向きのプロセッサが適している。

(2) 集合 *SOLID* (ソリッド形状) を対象とする機能モジュール: 写像 *R*

①処理: 直線とソリッド形状との相貫を与える写像 $R: SOLID \times RAYS \rightarrow SOLID$ で定義される。すなわちある3次元ソリッド形状 $A \in SOLID$, および3次元空間直線 $\alpha \in RAYS$ に対し、式(12)により、形状 *A* に含まれる直線 α 上の点の集合を与える。

$$R(A, \alpha) = \{x; x \in A \cap \alpha\}. \quad (12)$$

②ハードウェア: CADシステムで扱う形状をすべて扱う必要性から、形状との相貫計算において少なくともCADシステム側で可能な計算処理が実現可能である必要がある。この目的にはDSP(デジタル信号処理用プロセッサ)などの汎用数値計算向きプロセッサが適している。

(3) 集合 *SETP* (CSG形状集合演算式) を対象とする機能モジュール: 写像 *F*

①処理: 形状集合演算式、すなわち式(4)の *f* に等しい写像 $F: SOLID^{Np} \rightarrow SOLID$ で定義される。

CSGモデルの性質⁴⁾から、写像 *F* には次の性質が成り立つ。

(性質F) $S_i \in SOLID, i \in \{1, 2, \dots, Np\}$ に対して、

$$F(S_1, S_2, \dots, S_{Np})^b \subset \bigcup_{i=1}^{Np} S_i^b. \quad (13)$$

すなわちプリミティブ形状の集合演算の結果得られる形状の境界は、個々のプリミティブ形状の境界の和集合に含まれる。

②ハードウェア: 後の3.2.3節に示すように、*F*の処理は最終的に1次元上の区間に対する集合演算に帰着する。これは区間を表す情報の比較・交換操作にもとづいた簡単なハードウェアにより実現できることが示されている⁵⁾。(処理方式、ハードウェア構成等は文献5)に詳述する。)

(4) 集合 *PICTURE* を構成する集合 *COLOR* (色の集合) を対象とする機能モジュール: 写像 *C*

①処理: 3次元ソリッド形状の境界面上に位置する点での色情報を与える写像 $C: SOLID^a \times E^3 \rightarrow COLOR$. すなわち、 $S_i \in SOLID$ として、

$$S = (S_1, S_2, \dots, S_n) \in SOLID^a, \quad (14)$$

と定義すると、 $x \in S_i^b, i \in \{1, 2, \dots, n\}$ なる *x* に対して、 $C(S, x)$ によって S_i 上の点 *x* における色を与える。したがって *x* がいずれかの S_i の境界に属している次の条件が必要とされる。

$$(\text{条件C}) \quad x \in \bigcup_{i=1}^n S_i^b. \quad (15)$$

光源や反射モデルなどの、色を求めるための情報⁶⁾は、写像 *C* に含むものとする。

また、写像 *C* には次の性質が成り立つ。

(性質C) $A = (A_1, \dots, A_m), B = (B_1, \dots, B_m), A_i, B_i \in SOLID$ に対して、

$$\bigcup_{i=1}^n A_i^b \subset \bigcup_{i=1}^m B_i^b, \quad (16)$$

の関係があるならば、 $x \in A_i^b$ なる *x* に対して、

$$C(A, x) = C(B, x), \quad (17)$$

が成立する。すなわち形状の集合 *B* の境界を一部含む形状の集合 *A* は、共通の境界上で同じ色をもつ。

②ハードウェア: 写像 *R* と同様な議論から、汎用数値計算向きプロセッサが適する。

(5) 写像 C の定義域 E^3 を処理対象とする機能モジュール: 写像 V

①処理: 与えられた形状から, 可視な部分集合を選択する写像 $V: SOLID \rightarrow E^3$. すなわち, $S \in SOLID$ に対して,

$$V(S) = \{x; x \in S, \text{visible}(x)\}. \quad (18)$$

ただし, $\text{visible}(x)$ とは x が可視な点であることを表す条件である. 一般には光線上の最も視線方向側に位置することで, 条件 visible が成立する.

②ハードウェア: 後の 3.2.3 節に示すように, V の処理は最終的に値の最小値を求める処理へ帰着される. したがって適当な探索向き計算機アーキテクチャの採用が適する.

3.2 問題 2: 写像の合成

3.2.1 前 提

CAD システムのマシントラフェースの目的で画面を用いる場合, 一般に, 屈折・反射・影などの現象を省略した画像で十分実用となる. ここでは光源から出発し形状表面で 1 回の反射の結果視点に届く光のみを考慮した画面作成方式を対象とする.

以下プリミティブ形状として $P_1 \sim P_{Np}$, 形状集合演算式として f を用いて定義 3 により定義される CSG 形状 S を対象とする.

3.2.2 合 成

写像 $DRAW$ の値域である $SCREEN$ の要素 $COL(x)$ は, 画素 x 位置に投影される形状表面の色である. したがって投影される形状上の点を $VIS(x) \in E^3$ で表現すると, $COL(x)$ は,

$$COL(x) = C(S, VIS(x)). \quad (19)$$

ここで性質 F から, CSG 定義形状の表面は, そのプリミティブ形状の表面の和集合に属する, すなわ

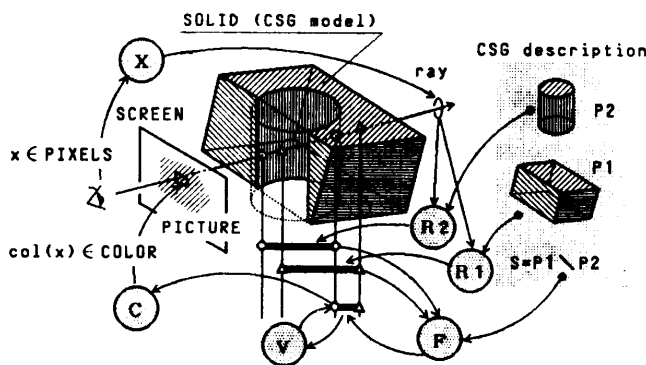


図 3 写像 $DRAW$ を合成する五つの写像の関係
Fig. 3 Relationship among 5 mappings which compose the mapping $DRAW$.

ち,

$$S^b = F(P_1, \dots, P_{Np})^b \subset \bigcup_{i=1}^{Np} P_i^b. \quad (20)$$

これに対し性質 C が適用できて, 式 (21) が成立する.

$$C(S, VIS(x)) = C((P_1, \dots, P_{Np}), VIS(x)). \quad (21)$$

すなわち色を計算する機能モジュールではプリミティブ形状の情報を知る必要がある.

次に $VIS(x)$ は, 写像 R, V, F, X を用いて, 次のように合成される.

$$VIS(x) = V(F(\{R(P_i, X(x)); i \in \{1, \dots, Np\}\})). \quad (22)$$

式 (22) は, 図 3 に示すように写像 X により生成した画素位置を通る直線と, Np 個のプリミティブ形状それぞれとの相貫区間を得て, それらに写像 F を適用して CSG 定義の形状集合演算を行うことで, 定義形状と直線との相貫区間を求め, 写像 V で, その 1 次元区間のなかから可視な 1 点を得るという, 一連の処理を示している. さらにプリミティブ形状を参照して可視な点に対する色を求めることで, CSG-CG 処理が実現される.

したがって, 式 (22) を用いて式 (19) を書き直して, 最終的に写像 $DRAW$ が上記の五つの写像を用いて次式で表される (図 3 参照).

$$PICTURE = (C((P_1, \dots, P_{Np}), V(F(\{R(P_i, X(x)); i \in \{1, \dots, Np\}\}))); x \in SCREEN) \quad (23)$$

3.2.3 機能モジュールの詳細の決定

式 (23) の導出は, 各写像の定義域・値域を一致させることを要求する. その結果, 機能モジュールの処理対象の範囲が次のようになる.

①写像 F の定義域 $SOLID^{Np}$ ・値域 $SOLID$ は, 1 次元の直線をなす $SOLID$ の部分集合へ制限され, したがって処理は 1 次元区間の集合演算へ帰着される. ②写像 R の定義域 $SOLID$ は, プリミティブ形状 P_i へ制限される. したがって式 (3) より, P_i を定義する実数値関数であるパターン関数 F_i を用いた数値計算処理へ帰着される. ③写像 V の定義域 $SOLID$ が 1 次元の集合に制限されるゆえ, その処理は 1 次元の集合上で最近点を見付けること, すなわち複数の区間の端点に対して最小値を決定する簡単な探索に帰着する. ④写像 C は, $SOLID^*$ が $SOLID^{Np}$ へ明確化される.

3.3 問題3：並列処理構成の導入

3.3.1 前 提

CSG 形状を定義するプリミティブ形状の個数 Np は定義する形状によって異なる。 Np をある値に固定して式のとおり R の機能モジュールを Np 個並べたことは、CSG 定義形状の任意性を損ねるから、ここでは一つの機能モジュールで Np 個のプリミティブに対する R の処理を行うことを前提とする。すなわち新たに写像 R を用意し、 $\{R(P_i, x); i \in \{1, \dots, Np\}\} \Rightarrow R(x)$ と置き換える。さらに写像 C の中に (P_1, \dots, P_{Np}) を含めて、式 (23) を次式に書き直す。

$$PICTURE = (C(V(F(R(X(x))))));$$

$$x \in SCREEN. \quad (24)$$

3.3.2 集合の分解による並列化の過程

写像 $DRAW$ を、それを定義する各写像の定義域・値域を部分集合へ分解することで書き換える。なお、ここでは任意個の並列処理構成へ分解する方法と、そのときの負荷分散の式を得ることを目的とする。最適な負荷分散のためのプロセッサ個数の関係の決定等は本稿では触れない。

いま写像 (機能モジュール) V, F を例にとって、 V を Nv 個に、 F を Nf 個に分解することを考える。 j 番目の写像 V である V_j の入力を J_j と表記する。

V_j と F_k 間の可能なすべての接続方法を含む接続方法として、ここでは図4に示すように、 V_j を F_k , $k=1 \sim Nf$ の出力のすべてと接続する方法をとる。そして、 V_j の入力 J_j は、 F_k それぞれの出力の V_j に関する部分集合 F_{jk} を、 k に関して和集合でまとめたものとする。すなわち、式 (25) なる関係式で与える。

$$J_j = \bigcup_{k=1}^{Nf} F_{jk}. \quad (25)$$

また、 F_{jk} は、 F_k への入力 K_k の中の V_j に関する

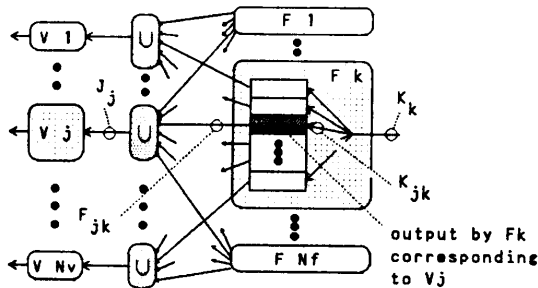


図4 プロセッサ V と F の結合方法

Fig. 4 Processor interconnection between V and F .

る部分集合 K_{jk} から、写像 F_k を適用して与えられるから、

$$F_{jk} = F_k(K_{jk}), \quad (26)$$

さらに、 K_{jk} は K_k の Nv 個の分類であるから、

$$\bigcup_{j=1}^{Nv} K_{jk} = K_k. \quad (27)$$

以上の操作を同様に $DRAW$ 中のすべての写像に対して繰り返し適用して、並列構成の式を以下のようを得る。

$$PICTURE = \bigcup_{i=1}^{Nc} C_i(I_i), \quad (28)$$

$$I_i = \bigcup_{j=1}^{Nv} V_j(J_{ij}), \quad \bigcup_{i=1}^{Nc} J_{ij} = J_j, \quad (29)$$

$$J_j = \bigcup_{k=1}^{Nf} F_k(K_{jk}), \quad \bigcup_{j=1}^{Nv} K_{jk} = K_k, \quad (30)$$

$$K_k = \bigcup_{l=1}^{Nr} R_l(L_{kl}), \quad \bigcup_{k=1}^{Nf} L_{kl} = L_l, \quad (31)$$

$$L_l = \bigcup_{m=1}^{Nx} X_m(M_{lm}), \quad \bigcup_{l=1}^{Nf} M_{lm} = M_m, \quad (32)$$

$$\bigcup_{m=1}^{Nx} M_m = PIXELS. \quad (33)$$

ここで、 Nc, Nv, Nf, Nr, Nx は、それぞれ写像 C, V, F, R, X に関する定義域の分割数、すなわち写像 (プロセッサ) の個数を表す。

次に、分割した各写像の値域を次のようにして定式化する。

写像 C を例にとる。 C が合成写像の導出により、その定義域が制限されて、

$$C: POINTS \times SOLID^{Np} \rightarrow COLOR, \quad (34)$$

と定まるとする。ここで、 $POINTS$ と記述する列の要素は、画素全体の列である $SCREEN$ の要素と 1 対 1 に対応する。そこで C の i 番目の分割、

$$C_i: POINTS_i \times SOLID^{Np} \rightarrow COLOR, \quad (35)$$

に対し、その定義域 $POINTS_i$ に対して、対応する画素の列の部分列を用意し、 G_{ci} で表現する。

$DRAW$ を合成するすべての写像において、その定義域は $SCREEN$ の列と対応するから、同様にその分割を定める定義域を、画素の部分列に表現して、それぞれ C_i, V_j, F_k, R_l, X_m に対し、 $G_{ci}, G_{vj}, G_{fk}, G_{rl}, G_{xm}$ で記述する。すると式 (28) ~ (33) は、 $G_{ci} \sim G_{xm}$ に対する関係式となる。これらの連立式を満足する $G_{ci} \sim G_{xm}$ として負荷の分散を求めることができる。

3.4 問題4：プロセッサ結合のハードウェア機構の決定

負荷の割り当て方法の1方法を述べ、そのハードウェアの機構を示す。以下の議論は機能モジュールCとVを例にとるが、同様な議論は他の写像間の接続にも成り立つ。

目的とする画素の部分列の要素が、元の画素列に均等に分散していることを目的として、以下の式を負荷の分散の解として用意する。

$$Gc_i = \{PIX_k; (k-1) \bmod Nc = (i-1), k \in \{1, \dots, Ns\}, PIX_k \in PIXELS\}, \quad (36)$$

$$Gv_j = \{PIX_k; (k-1) \bmod Nv = (j-1), k \in \{1, \dots, Ns\}, PIX_k \in PIXELS\}, \quad (37)$$

すなわち全画素列の要素をプロセッサC、Vのそれぞれの個数の剰余で分類して割り当てる方法を示している。

この場合、

$$\bigcup_{i=1}^{Nc} Gc_i = \bigcup_{j=1}^{Nv} Gv_j = PIXELS, \quad (38)$$

が成り立つゆえ、ハードウェアはこの式のとおり、Gv_jに属する機能モジュールの出力を和集合でいったん元のSCREENの列へ並べ直して、逆にGc_iに対応する機能モジュールへ必要な部分を配分し直す方法をとる。このハードウェア機構として、図5に示す機構を用意する。

すなわちNv個のバスから1入力を選ぶマルチプレクサを用意し、1画素に対応する出力要求ごとに順にNv個のバスを選択・転送する動作を行うことで、式(38)のNvに関する和集合演算を実現する。同様に1入力Nc出力のデマルチプレクサを用意して、その出力先Ciの、次の出力先であるC_(i+1 mod Nc)が入力可能になった時点で切り替えを行い、入力情報を流すことで、式(38)のNcに関する和集合を実現する。図

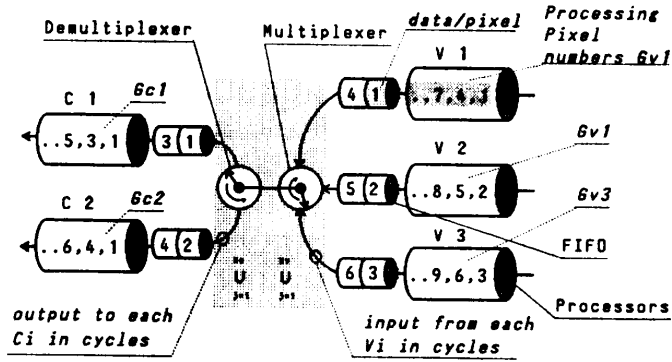


図5 プロセッサ結合のハードウェア機構
Fig. 5 An example of the schematic hardware architecture of a processor interconnection.

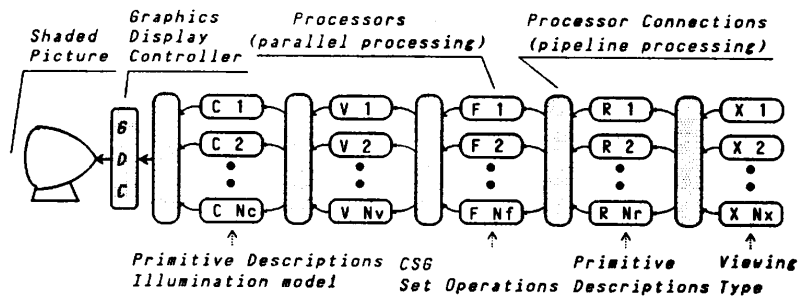


図6 CSG-CG専用機のプロセッサ構成
Fig. 6 An organization of the specialized processor system for the CSG-CG process.

5は、Nv=3、Nc=2のときの動作状況を示している。

3.5 CSG-CG専用機の全体構成

以上より、非同期パイプライン構成のCSG-CG専用機が図6の構成で得られた。図中GDCと記述するユニットは、Ciの出力に対する和集合操作に相当するハードウェアで、各Ciの画素列を格納するローカルなフレームバッファと、それを合成して表示する部分よりなる。(ここでは詳細は省略する。)

4. おわりに

本論文では、CSGモデルからの濃淡面画作成処理を専用ハードウェアプロセッサとして実現する方法の定式化について述べた。以上の結果は以下のようにまとめることができる。

- ① CSG-CG処理を集合・写像の枠組で定式化した。
- ② 並列処理を含めたCSG-CG専用機的设计問題を、写像の合成問題として扱う枠組を示した。
- ③ 上記の枠組の正当性を検証する目的で、非同期

パイプライン構成の専用プロセッサを設計する例を示した。

参 考 文 献

- 1) 沖野教郎: 自動設計の方法論, 養賢堂 (1982).
- 2) 沖野教郎: 新ソリッドモデラのための CSG 再考, 精密工学会誌, Vol. 53, No. 3, pp. 357-360 (1987).
- 3) 元岡 達編: VLSI コンピュータ II, 岩波書店 (1985).
- 4) Voelcker, H.B. and Requicha, A.A.G.: Geometric Modeling of Mechanical Parts and Processes, *Computer*, Vol. 10, pp. 48-57 (1977).
- 5) 三上貞芳, 内堀勝章, 嘉数侑昇, 沖野教郎: CSG グラフィック・エンジンの開発, 精密工学会昭和63年度春季大会講演論文集, pp. 473-474 (1988).
- 6) Rogers, D.F.: *Procedural Elements for Computer Graphics*, McGraw-Hill (1985).

(昭和63年11月21日受付)

(平成元年6月13日採録)



三上 貞芳 (正会員)

昭和37年生。現在北海道大学大学院工学研究科精密工学専攻博士後期課程在学中。ソリッド形状モデリング, 機械設計 CAD/CAM システム, グラフィックス, 専用ハードウェア等に興味をもつ。日本機械学会, 精密工学会各会員。



嘉数 侑昇 (正会員)

昭和16年生。昭和48年北海道大学大学院工学研究科精密工学専攻博士課程修了。工学博士。北海道大学工学部精密工学科教授。CAD/CAM, 画像処理工学, ロボティクス, 知識工学, ニューラルネットワーク等の研究に従事。日本機械学会, 精密工学会, 日本ロボット学会等の会員。