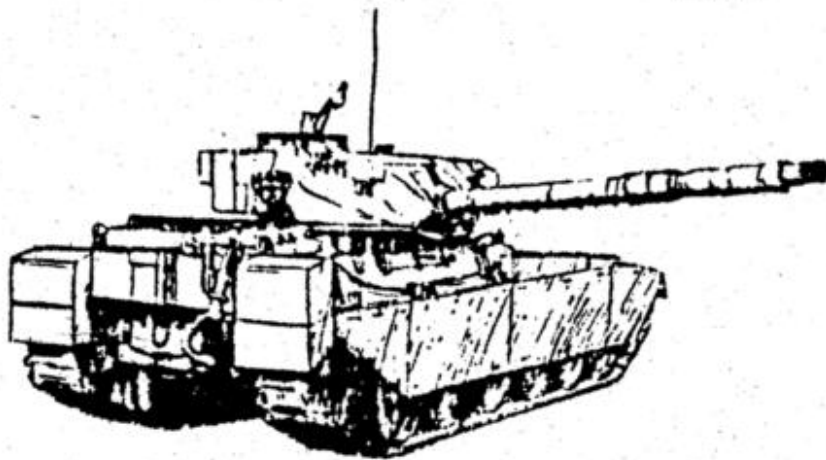


Lime



K.I.T. Computer
Club

1986.
学祭号

No.10

1986年 第 4 号 特集 特集 特集

目次

- 3 P. レイトレーシングの原理
小林 弘樹, 田中 辰巳
- 19 P. 戦場からマイクロマウスへ
田中 照寛
- 21 P. OBの部屋 Tom さんの近況報告
Tom.
- 23 P. (i) UNIX が 大嫌 い !
- 28 P. (ii) BSD v.s. System V
- 29 P. (iii) コンピュータで飯を喰べたい方へ
Take♡
- 31 P. 編集後記
増田 隆夫

レイ・トレーシングの原理

コンピュータグラフィックスといえば、物体の骨組みを線で表したワイヤフレームモデルが思い出されると思うが、反射や屈折をも考慮したリアルな3次元グラフィックスが商業などで実用化されたのは比較的最近のことです。このようなグラフィックスを描く方法の主流がレイ・トレーシングです。表現に富み、それだけに大変な計算を要しますが、アルゴリズムそのものは単純です。よって計算時間を無視すれば、パーソナルコンピュータでも容易に実現できます。以下にレイ・トレーシングの原理を説明します。

1) 光線の追跡

まず乱反射(不透明)物体、鏡面体、透明体が図1のように並んでいる状況を考えましょう。光源から出た光は、物体によって

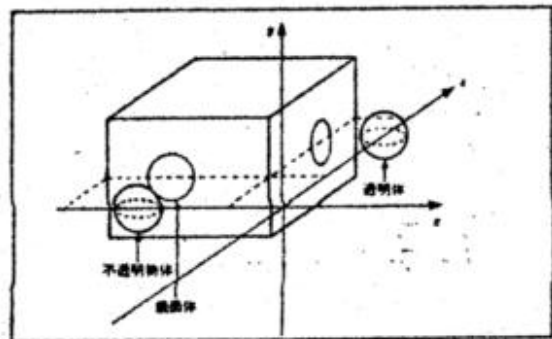


図 1

乱反射、鏡面反射、あるいは屈折して目(視点)に戻ります。水晶体や網膜のことは忘れておくにすれば、これが「ものを見る」ということなのです。

レイ・トレーシングでは、この過程をコンピュータにおいてシミュレートし、リアルな3次元画像を得ることになります。もう少し詳しく言うと、この過程をシミュレートすることによって

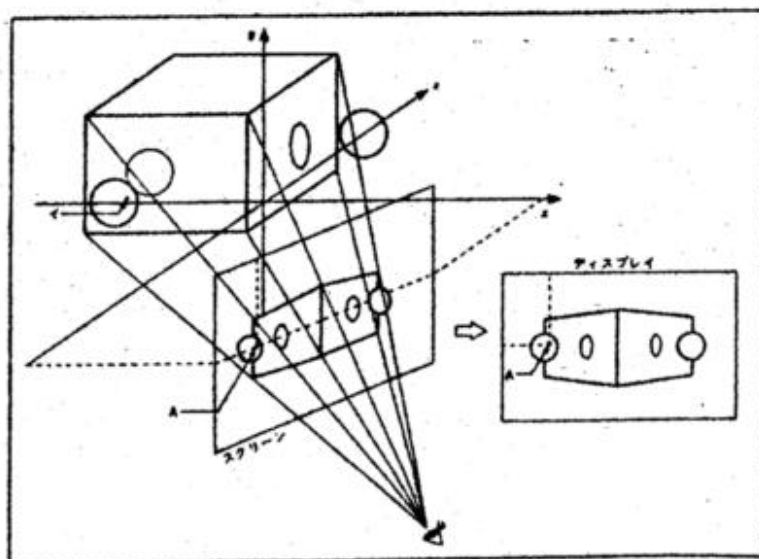


図 2

画面上のすべての画素に適切な色、明るさをつけるというこ
 なのです。図2左のように視点と物体の間にスクリーンを
 置きましょう。まず、ディスプレイ画面上のある画素Aに対応
 するスクリーン上の点Aから視点に届く光の色や明るさ(以
 下、成分と呼ぶ)を計算します。次にディスプレイ画面の
 画素Aにそれに応じた色をつけます。これを画面上のすべ
 ての画素について繰り返せば、画面にはちょうど視点から
 スクリーンを通して物体をながめたのと同じ絵が描かれる
 わけです。(図2右)

光の成分を計算するとき、視点とスクリーン上の点Aとを結ぶ
 進路(視線)上を逆方向にたどってこの光がどの物体から
 や、またのかを探る必要があります。これがレイ・トレーシ
 ング法、光線追跡法、あるいは視線探索法と呼ばれる
 ゆえんです。

2) 散乱光, 乱反射光, ハイライト.

では, 図2の点Aを通, 2で視点, に好てくる光線の成分の計算から見てみたいと思います. 視点からA点に向う視線をたどっていくと, 点Aで不透明物体にひつかります. 不透明な乱反射する表面をもつ物体からの光は 散乱光, 乱反射光, ハイライトの3つからなっています.

いま, 机の下など 陰になっているところをみてみると, Σ はたとえ照明の陰であっても真, 暗ではなく, ちゃんと物が見えます. これは光源からの光が壁や床や空気中のちりなどによって反射され, あらゆる方向に向いて あらゆる場所を照らしているからです. これを散乱光といいます.

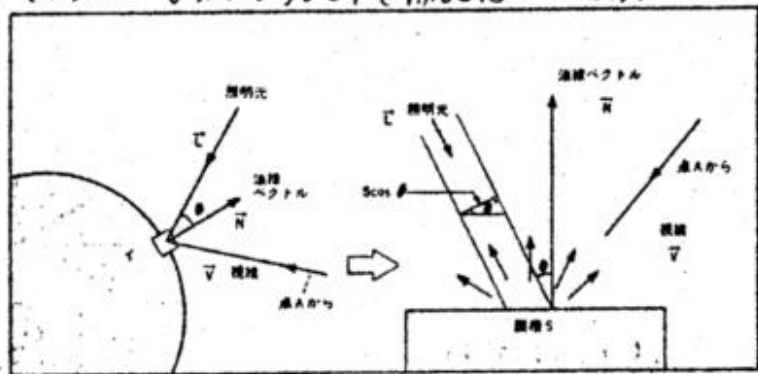


図3

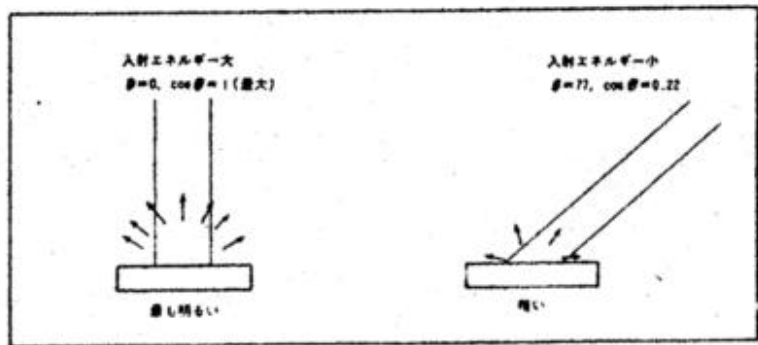


図4

6

乱反射光は照明の
当たり具合によって変
化が明瞭なことを表し
ます。

点イでは照明光が
乱反射、つまりあらゆる
方向に等しく反射
されます(図3)。した
がいて視線の方向
は関係ありません。

ある方向に反射され
るエネルギーは点イ
(の周りに小さな面積 S)
に注がれる照明光の
エネルギーに比例し
ます。照明光の方向と点イ
の物体表面に垂直な
方向(法線ベクトル)
となす角を θ とする

と、入射する照明光のエネルギーはそれを垂直に切った面積

$$S \cdot \cos \theta$$

に比例します。結局、乱反射光は $\cos \theta$ に比例するわけ
です。表面に垂直に光が当たっているとき、 θ が 90° の最も明るく、水平
に近いほど暗くなるというのはよくあることです(図4)。

散乱光と乱反射光だけでは物体はまったくのつや消して、現実
味に欠けます。そこで先ほどは無関係であった視線の方向も
関係するような反射を考えます。これをハイライトといい、よくイ
ラストなどで球を表わすのに使われています(図5)。ハイライトは
物体表面と照明光が鏡面反射し、その反射光の見え具合に

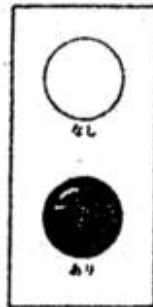


図 5

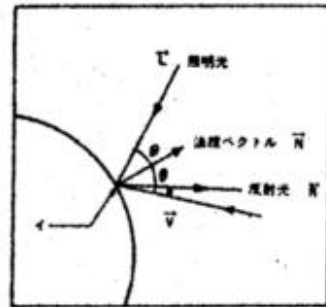


図 6

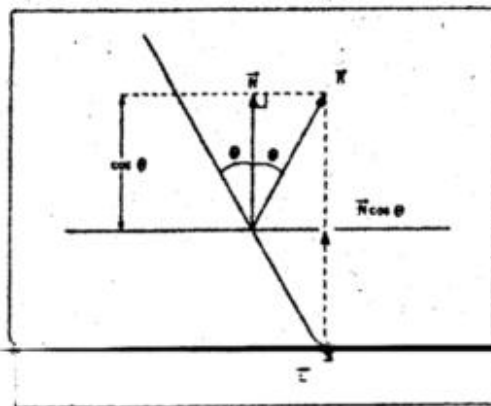


図 7

よって現れると考へます。図6はその様子を示しています。
 反射光がまともに目に入るほど、つまり反射光と視線の
 なる角が小さいほど、ハイライトを大きくさせます。このた
 めに、ハイライトの強さは $\cos \theta$ の n 乗に比例するにします。
 $\cos \theta$ の値を計算するのに、まず反射光の方向 \vec{R} を求め
 ましょう。図7から明らかのように、

$$\vec{R} = \vec{N} + 2\vec{N} \cos \theta$$

と書けます。視線の方向ベクトル \vec{V} とすれば θ の値は

$$\text{内積 } |\langle \vec{R}, \vec{V} \rangle|$$

で計算できます。

乱反射物体の表面の点 P から点 A を通って視点に至る
 光線のエネルギー E は次のように求まります。

$$E_p = E_d + A \cos \theta + B \cdot |(\vec{L} + 2\vec{N} \cos \theta), \vec{V}| n \dots \textcircled{1}$$

$$\cos \theta = |(\vec{L}, \vec{N})|$$

ここで、 E_d は定数、 A と B はそれぞれ物体に固有な比
 例係数。 \vec{L} は照明光の方向、 \vec{N} は表面の法線ベクトル、
 \vec{V} は視線の方向です。

これまでの光線のエネルギーについては明瞭では分かっては
 分かりません。つまり白黒です。どんな色も赤 (red)、青 (blue)、
 緑 (green) の三原色によって表わせるので、光線の r, b, g の
 各成分について、エネルギー計算を別々に行い、その結果を重ね
 合わせてやります。このとき、 r のエネルギーを計算するときの $\textcircled{1}$
 の係数 A の値 A_r と g, b のそれぞれ A_g, A_b の値は一般
 に異なります。係数 B についても同様です。これがつまりそ
 の物体の色の違いとなって現れるのです。

3) 鏡面体と光線

次に図8に示したスクリーン上の点 B を通って視点に至る光
 線について考えましょう。点 A のときと同様に視線をたると

8

今度は点
で鏡面体
にのりか
ります。
鏡面体上
の点から
視点にや
ってくる
光線は2
つの要素
に分けて
考えられ
ます。1つ
は鏡面反
射による
もの。う
つは点の
場合と同
じ散乱光
、乱反射
光、ハイ
ライトに
よるもの
です。
図中に示
した

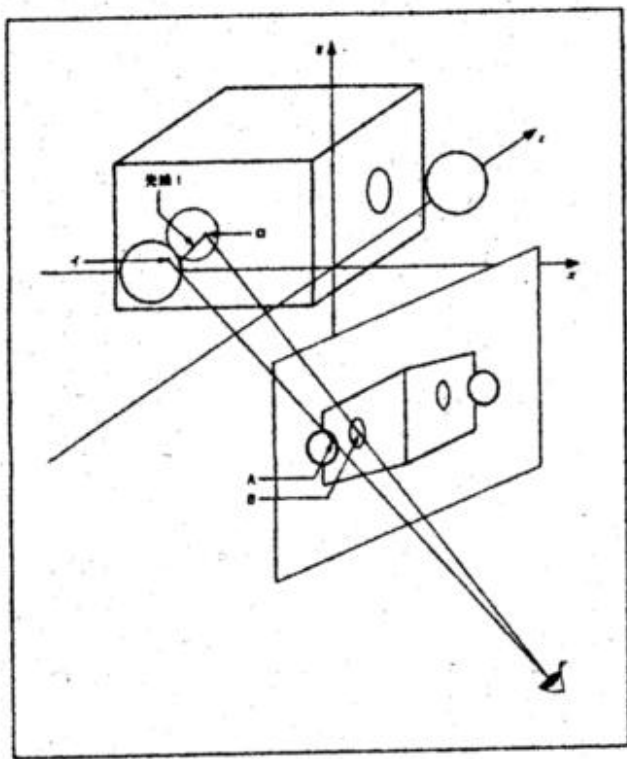


図 8.

光線1は点
の鏡面反射して点Bを通り、視点に届きます。したがって点Bか
ら視点にくる光線の成分を計算するためには、光線1の成分を

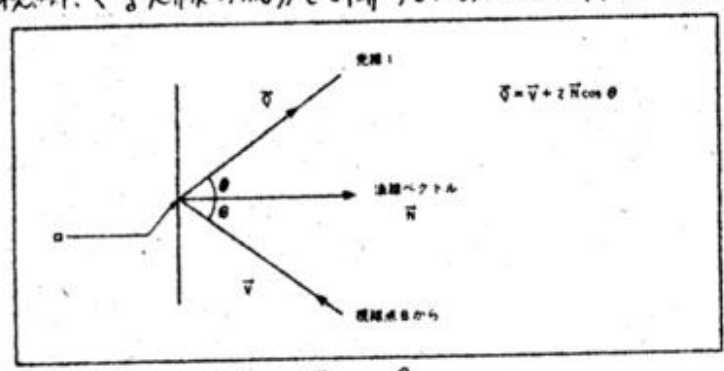


図 9.

$$\vec{v} = \vec{v} + 2\vec{N}\cos\theta$$

知らなければなりません。そこで視線が点Oで反対⁹された方向をまず考え、光線1をたどっていくわけです。図9にその方向を示します。図9とまったく同様ですから、視線の方向をV、点Oでの法線方向をN、反射方向をQとすると

$$Q = V + 2N \cos \theta \quad \text{--- ②}$$

$$\cos \theta = |(V, N)|$$

これで反射された視線の方向が分かり、また視線の始まる点も分かっています(点O)。

そこで元の視点の代わりに点Oを新たな視点と考えると、光線1の成分を計算するのに、ここで説明している方法をそのまま使うことができます。図8の例では反射した視線をたどると点Iと同じ球に当たります。したがって、先ほど説明した点Aからの光線と同じ方法で光線1の成分を計算することが出来ます。もし反射した視線がまた鏡面に当たるといけば、点Oで行ったようにその交点を新たな視点と考える。さらに同じ方法を適用するのであります。

さて、鏡面といっても完全なものではなく、鏡面に当たる光のうちいくらかはきちんと鏡面反射されずに勝手な方向に反射されます。つまり、光線1のうちいくらかは点Bに向かわないのです。逆に点Bを通る視点に届くのは光線1の何割かです。この割合を反射率といいます(図10)。鏡面に当たる光のうちいくらかは乱反射される

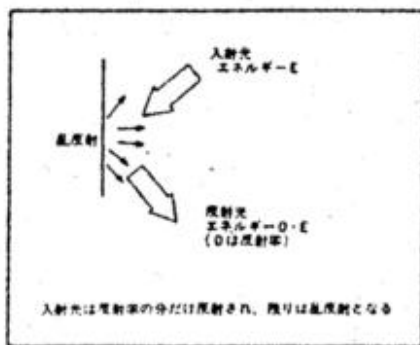


図 10.

10. のですから、この分については点Aで考えたのと同じ式①が使えます。また鏡面上の点Aから点Bを通って視点に至る光線のエネルギーは次のようにまとめられています。

$$E_r = C \cdot E_p + D \cdot E' \quad \text{--- ②}$$

ここでC, D (反射率)は物体の固有な係数, E_p は①を参照, E' は反射方向からの光線のエネルギーです。

③は, r, g, b成分1つ1つについての式ですから, 実際の計算は3つの成分について行わなければなりません。

4) 透明体と光線

今度は図11の点Cを通って視点に届く光線の成分を計算します。視点から点Cに向かっ
て視線をたどると点Aで透明体に当たります。ここでは透明体は無色とします。透明体上の点での光線の計算は鏡面の場合よりさらに複雑です。乱反射による部分はありませんが

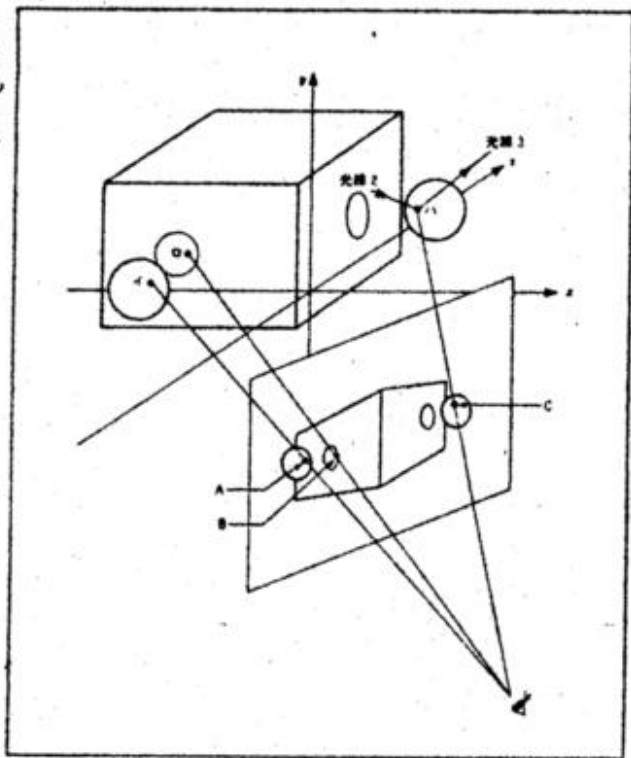


図 11.

鏡面反射によるもの, 屈折によるものが関係してきます。透明体は光を屈折させて通すと同時に反射もするので。

点ハにおける光の
 屈折を書くと
 図12の
 ようになり
 ます。光線2
 は点ハで
 反射して点
 Cに向かう
 光線、光線

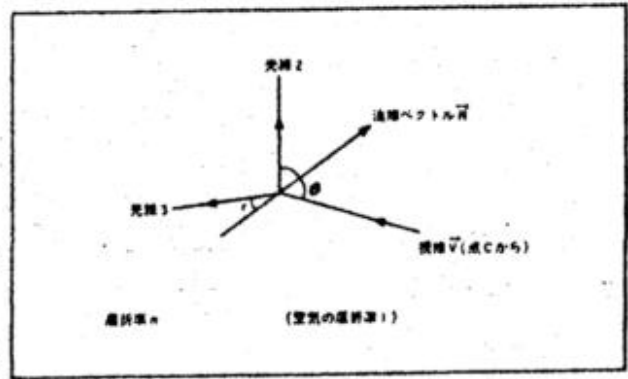


図 12.

3は点ハで屈折し
 て点Cに向かう光
 線です。点ハ、点Cを
 通る視線に届く光
 線の成分を計算す
 るには、やはり光線2,
 3を逆にたどってその
 成分を計算するけ
 ればなりません。光
 線2については矢の
 鏡面反射と同じく
 同じように計算でき
 ます。

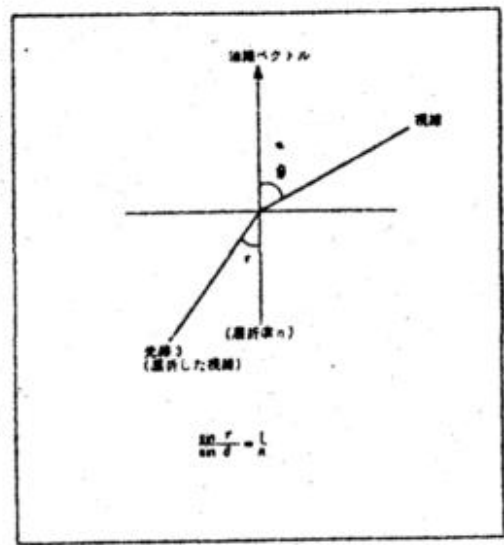


図 13.

では光線3とたどる視線の方向を求めましょう。2次元の
 場合の屈折の方向はよく知られたスネルの法則で決定さ
 れます(図13)。図14も同じスネルの法則を表わしていま
 す。今度は一一般の3次元ベクトルを用いたものです。実
 際に計算してみると、2つの図が同じことを表わしているが

12 分かります。結局、屈折の方向Pは

$$P = k(\vec{V} + \vec{N}) - \vec{N}$$

$$V' = V / \cos \theta$$

$$= V / |\vec{V} \cdot \vec{N}|$$

$$k = \sqrt{n^2 |\vec{V}|^2 - |\vec{V}' \cdot \vec{N}|^2} \quad \text{--- (4)}$$

ここで、 \vec{V} は視線の方向、 \vec{N} は法線ベクトル、 n は屈折率、 θ は入射角を示します。ただし、 k の式のルートの中が負になったときは光線が全反射するを示します。

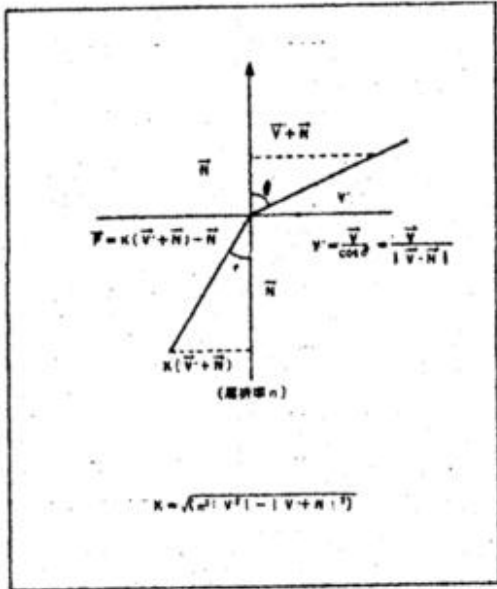


図 14.

これで屈折した視線の始点(点A)と方向が求められました。光線1, 2と同様に、点Aを新たな視点としていま説明しているが、自身を用いて光線3の成分を計算する事ができます。光線2, 3のすべてのエネルギーが点Aから点Cを通って視点に至るわけはありません。光線2のエネルギーが点Aで反射される割合を反射率、光線3が点Aを通りぬける割合を透過率といいます。この2つの値は次のフレネルの法則で求められます(図15)。

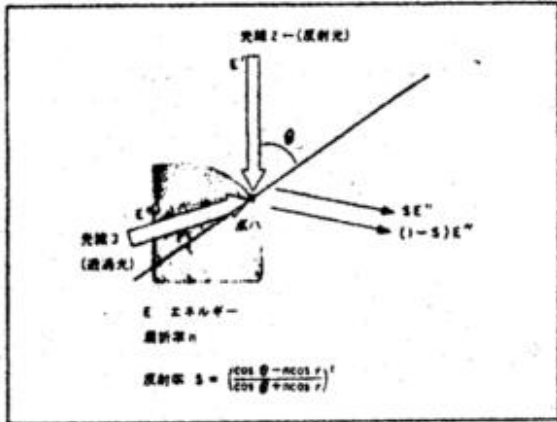


図 15.

反射率 S は

$$S = \frac{(\cos\theta - n \cos r)^2}{(\cos\theta + n \cos r)^2}$$

透過率 T は

$$T = 1 - S \quad \text{--- ⑤}$$

ここで θ は入射角、 r は屈折角、 n は屈折率を示します。なお、この式は正確な式を簡単にしたものです。

以上から、透明体上の点 A から点 C を通って視点に届く光線のエネルギーは次のようにまとめられます。

$$E_c = SE' + (1-S)E'' \quad \text{--- ⑥}$$

S は反射率、 E' は反対方向からの光線のエネルギー、 E'' は屈折方向からの光線のエネルギーです。

鏡面反射のときと同じく、今度も視点に届いたときのエネルギーがある限度以下の光線については計算を打ち切り、視線を2つに分割する回数の上限を決めておく必要があります。というのは、いったん透明体に入った視線はいつでもその内

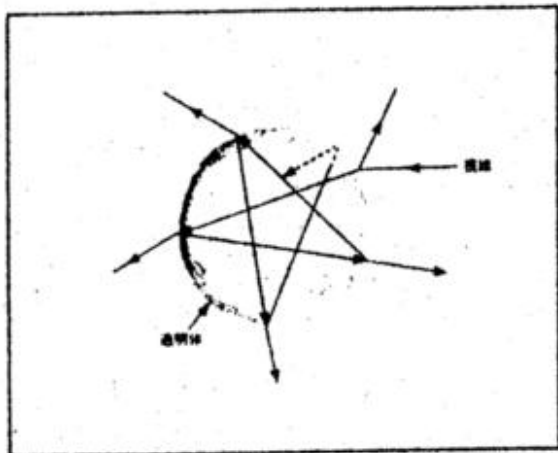


図 16.

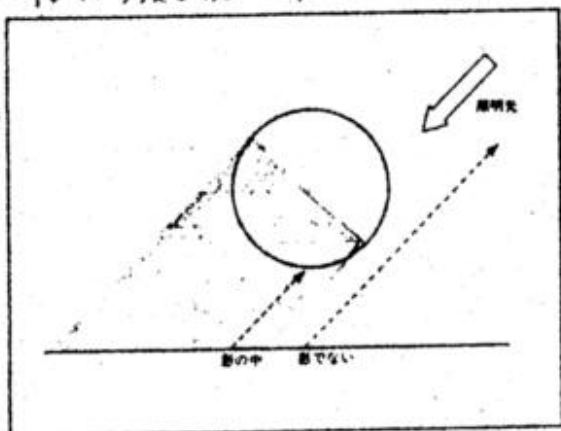


図 17.

14 部の反射を続けるからです(図16)。視線が透明体の外に出ると
 まは屈折率を

$\frac{1}{n}$
 と考えれば、または視線が透明体に入るとおおよそ同様です。

5) 影の中にある場合

これはこの話では視線をたどった先の点、たとえば点イには、
 常に照明が当たり、ているものとしてきました。しかし点イが何かの
 影の中にある場合もあるわけです。これを調べるのにも光線追跡
 を行います。点イを新たな視点と見て、光線の方に向かう視
 線をたどります。もし途中で何かに向かれば、点イはその物体
 によって照明光を遮られていて、つまり影の中にあるといえます
 (図17)。照明光が当たっていない場合でも散乱光によって照らされて
 いるので、点イから視点に届く光線のエネルギーは0にはなりま
 せん。

6) 物体と視点との交点を求める。

レイ・トレーシングでは画面上のすべての画素に対して対応する
 スクリーン上の点を、さらに視点からその点に向かて視線を
 たどり物体との交点を求めます。そして、その方向から視点、に

やってくる光線
 の成分を計算
 し元の画素に
 色をつけるこ
 が基本です。
 このうち計算
 の中心は、視
 線と物体との
 交点の計算と
 交点からの光
 線の成分計
 算です。後者

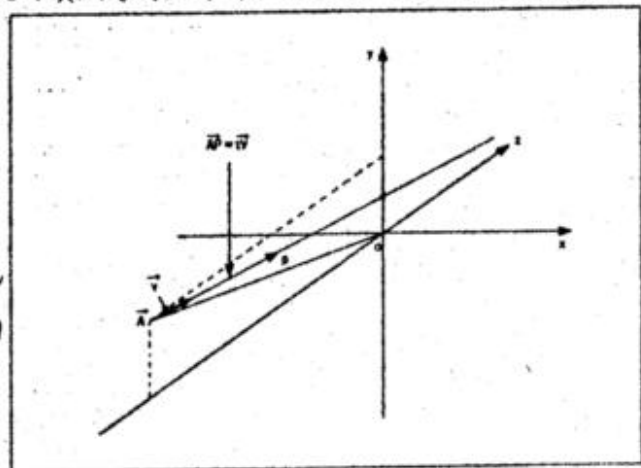


図 18

については、いま説明したとこです。前者について考え直してみよう。複雑な3次元物体をできるだけ単純な形の組み合わせとして表現しようとする時に、組み合わせの素材となる単純な形のものをプリミティブと言います。この時、そのプリミティブのAND(2つ以上のプリミティブどうし重なり合っている時の共有部分)や、OR(1つ以上のプリミティブのどちらか一方にでも含まれているすべての部分)を考慮なければ、それはいたって簡単になります。

x, y, z をプリミティブの中心を原点とする3次元座標の値とします。ある点 (x, y, z) がプリミティブの内部にあるといふことを数式

$$f(x, y, z) < 0$$

というときは同じことを表し、逆に、

$$f(x, y, z) > 0$$

と、点 (x, y, z) がプリミティブの外部にあるといふことは、同値である。さらに、

$$f(x, y, z) = 0$$

のとき、点 (x, y, z) はちょうどプリミティブの表面にあります。

7) プリミティブのデータ化

さて、プリミティブが数式で表わされるということが分かったので、次に視線を数式で表します。図18を見ると、原点を O 、視点 A 、 V は視線の方向を示すベクトルです。視点から視線上の点 P に向かうベクトル AP は V と同じ方向ですから、適当な数 t によって

$$\vec{AP} = t\vec{V}$$

と書けます。よって P の位置ベクトル \vec{OP} は

$$\vec{OP} = \vec{OA} + \vec{AP} = \vec{OA} + t\vec{V}$$

です。逆に、この式の t を変化させれば、点 P は視線上で動くこととなります。このとき、 \vec{V} の大きさを1にしておけば、 t は視点と点 P の間の距離を表わします。視点を (O_x, O_y, O_z) とし、

16. 上のベクトル式を成分に分けて書けば、視線上の点 (x, y, z) は

$$x = O_x + \lambda V_x$$

$$y = O_y + \lambda V_y \quad \text{--- ⑥}$$

$$z = O_z + \lambda V_z$$

となります。視線とプリミティブの交点は視線上にあり、かつプリミティブの表面上の点です。よって、交点 (x, y, z) では次の式が成り立ちます。

$$f(x, y, z) = f(O_x + \lambda V_x, O_y + \lambda V_y, O_z + \lambda V_z)$$

$$= g(\lambda)$$

$$= 0$$

この式から λ を求め、さらに交点 (x, y, z) を求めれば済みます。

次に「プリミティブのANDをとったもの」と視線の交点を求めることを考えます。図19の例を考えましょう。プリミティブA, B, CのANDをとったものは、「Aの内部であり、Bの内部であり、かつ、Cの内部である部分」であるから、まずプリミティブAと視線との交点を求めます。ここで交点をもたないときは分かれば、このANDをとったもの」と視線は交点をもたずありません。ここで交点を求めます。ただし、プリミティブの内側と外側を表す極性が-1な

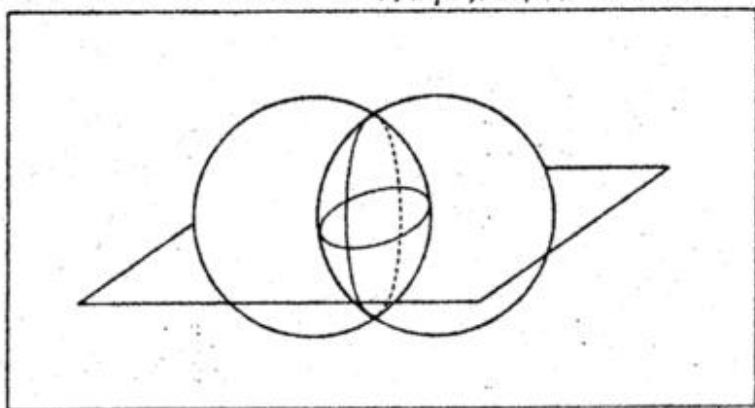
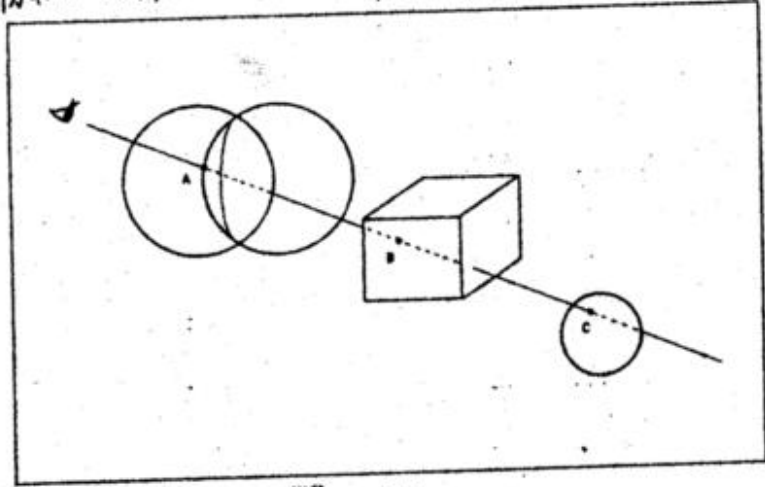


図 19.

らは、内側と外側が逆なので、次に進みます。交点があった場合は、その交点が B, C の内部に入っているかどうか調べます。どちらか一方でも入っていない場合は、この交点は「AND」としてのものの中にはありません。そこで、今度は B と視線の交点を求めます。交点が存在しなければなりません。交点が A, C の内部に入っていない場合はさらに C と視線の交点を求めます。この交点が存在し、かつ A, B の内部にあるならば、それが求める交点です。

要するに、1 つのプリミティブと視線の交点を求め、それがほかの AND としているプリミティブの内部にあることを確かめればよいのです。ただし、この条件を満たす交点が 1 つ以上存在する場合もありますが、そのときはこの中で最も視点に近い（最も小さい値をもつ）交点です。

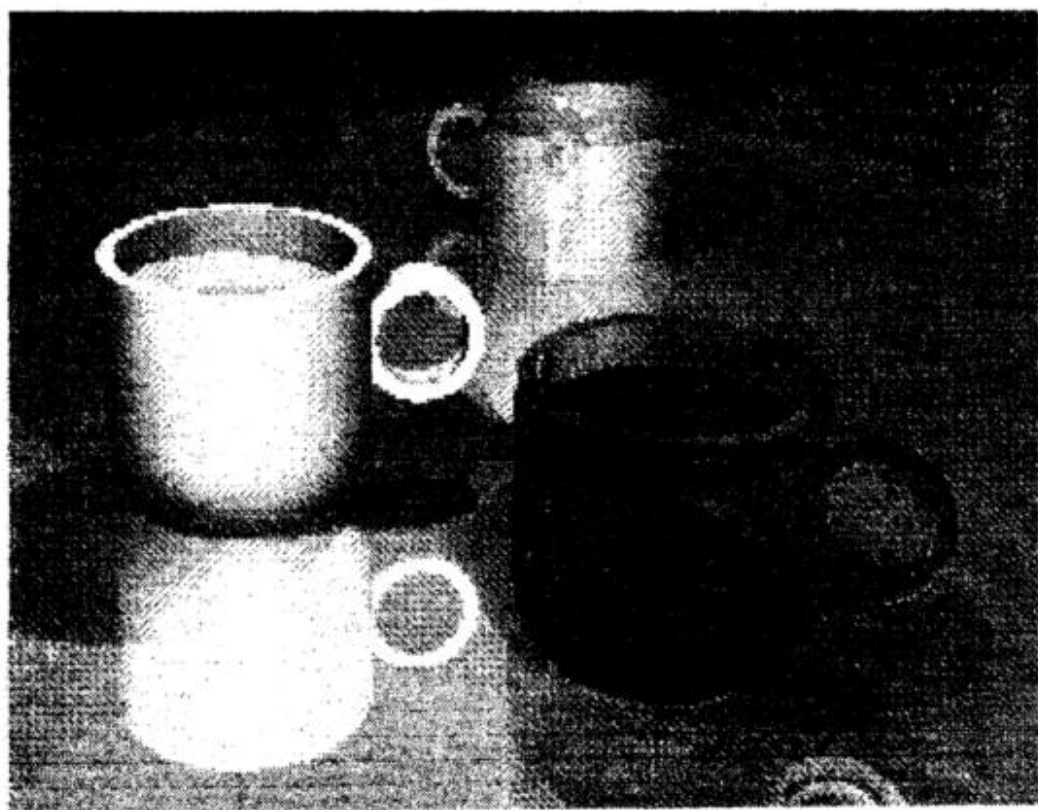
OK、この話は簡単です。視線はいつかのプリミティブや「AND」としてのものと交点をもつわけですが、その中で視線が実際に通る方向の視線を視点からたどったとき、いちばん最初に物体に当たる点です。つまり、最も視点に近い交点



18.

が求まる交点です(図20).

以上でライトレーシングの基本的な考え方の説明を終ります。



「戦車からマイクロマウスへ」

田中照憲

「マイクロマウスを作ろう」ということからこの企画は生まれました。が、マイクロマウスについての知識を詳しく持っている人はほとんどいません。短い期間で誰もが頭に浮かんでいるような M.M. を完成させるのは当然無理です。したがって迷路をさまよう複雑な動作はできなくても、単純である程度自立の動作をするものを製作することにします。私は M.M. の外面、駆動部の製作を主に任せられました。1年前プラモデルの走行をパソコンで無線誘導するという企画をして結局送信器のコンデンサを派手にはけさせただけで終わったのですが、私は今回も M.M. のシャーシにプラモデルを使うことにしました。このような企画にプラモデルを使う長所は、確実に動く駆動部を手軽に入手できるということです。しかしプラモデルは既に

大きさが確定しているものを入り切る故に搭載重量、搭載姿勢に制限があります。その点を考えて今回使用するものにしたのは、 $1/25$ スケールの戦車です。全長は30cm位ありますが、ギャボックス・バッテリー・基板がなるとか細まる程度です。2つのモータを使って前進、後進、回転の動作を行う戦車は今回の企画に最適と思われます。このプラモデルは付属のリモコンボックスのリバーを人間が操作して各動作を行います。今回はこの操作をリレーを使って行うこととなります。マイコンを載せてあらかじめプログラムした行動パターンに従い戦車が動けば完成です。すでにM.M.の面影はピンにも無くなっていますが、それは今後の課題です。戦車の砲塔を回転させるとか各種センサーを載せるなど様々なアイデアが出ており奥の深い企画になるかもしれません。



Tomさんの近況報告

21

Tomです。現在 NECの基本ソフトウェア開発本部 データベース開発部 という
と3で オンラインデータベース関係の仕事をしています。NECの汎用コンピュー
(ミニコンを除く)のOSは ACOS-2(小型), ACOS-4(中型), ACOS-6(大型), UNIX(
いよいよ)の大きく分けて4種類があります。このほかに エンジンプログラミングス
テム用, ミニコン用, スーパーコンピューター用, ハイビット Fortranプロセッサ用 OS など
があります。私は この中で ACOS-4の担当です。本学にある ACOS
-850用のOSは ACOS-6 ですが, ACOS-4は ACOS-6とは かなり異なります。
私の所属している部では データベースの中身ではなく データベースの いれものを
作っています。といっても私自身の仕事はこの部の本流の仕事では
なく, 大学時代の知識と経験を生かした『変な仕事』です。

データベース管理システムの, OSへの新しいインプリメント方式を調べる
というのが おもな仕事でした。今は その結果に基づいて
プロトタイプを作っています。調べたことというのは マルチタスクの
管理, ロードモジュールのメモリーへのロード方式, PL/I コンパイラが出力
するオブジェクト, …… etc. で およそ データベースとは 直接関係が
ないように見える ことばかりです。

私にとっては おもしろかったのですが こういうことに耐えられる人
にとっては 何年間も 一人な仕事(けっこう忙しい)を続けるのは かなりの
苦痛でしょう。私と同期で基本ソフトに配属された人間で,
製品を作る仕事をしていながら また ユーザーが せっせと いない
のは 私だけのような気がします(同期は 院修, 学部卒,
高専卒, 短大卒, 専門学校卒の男女を合わせ約80% います)。

このため, ユーザーの へんくさ 関心なく いていいので 楽だとも

み直し、良き^口もくたれ田^田になつて 私と同期の人間を どど人
苦しめてください。

なお、私は 二の仕事をほかに、特許関係の雑用、木構造団
(SPD) 関係の雑用、プログラム解析ツール(AI?) 関係の雑用
も やっていました。

とこ3で、開発環境は ACOS-800, FACOM M-380, FACOM
M-382 を current user 2~3 で使っていた
項に比べると 非常に悪いので、もっと良い環境がほしい
といふは ACOS-4 関係には来ないほうがよいでしょう。

PS. 今年、大中研一&が ACOS-4 のネットワーク管理
(とろくさいよ ~) のとこ3 に来ましたが、私のとこ3
より忙しいみたいです。

1986.11.22. (Sat.)

13:50

(24才最後の日に記す. by Jom)

UNIXが大嫌い!

SUN, JUNET, PDP, etc etc ...

TakeB

(take@kaba)

☞ たくし、近頃は、SUN-3 などというものを使うはめに
おちいりまして、UNIXなどは、いやだよ〜。

☞ Window Systemなどに、おがれていましたが、SunViewは
はっきりに不便、Mouseに持ち替えるのも、めんどくさい。
で、皆さんは X-Window を使うのね。(X-WindowはMITで
つくられた Systemで、「SONYのNEWS」では標準装備)
ところが私などは、もうめんどくさくて、VT-125から Sun-3へ
Loginしちゃう。こうなると、Sunも、おに VAXの、のりだね。

☞ いたし、Sunなど、「rlogin (Remote Login)」など、いわして
自分の Sun から、他の Sun へ 入って、そっちの CPU を使える。
そして、自分の CPU は、単なるターミナル・エミュレータになって(き)。

ところが、Sun-2と、いろいろが、また違い、昔は使えたらしいが、O.Sのリリースが、変って、どうしようもなくなった(真にIBM商法だね)。で、これを、皆さきは、Sun-2 = login (たと、同時に、Sun-3の rlogin する。今では、Sun-2は「rogue マシン」と呼ばれている。(「rogue」は、アドベンチャーゲームの名称)

わたくしは、UNIXは、ほとんど「わからず」、困っているのだが、やはり、UNIXは、「ええかげん」な、Systemやね。いっしょに働いている奴らは、やたら UNIXの中身に詳しいので、色々、尋ねるが、どうも、ええかげんで、BSDなど、「やはり、学生が、付けた機能やね」と、思うことが、たくさんある。で、わたくしなどは、

PDP-11などを持っている (正確には Professional-350を無限に借りている)ものだから、つい、DECの native O.Sで、おの RSX-11 (RSX-11M+) と、UNIXを比べてしまう。

すると、やはり、DECの方が、美しいのである。若干の古さも、感じさるが(生立ちが古いので)、ちやとにしているのである。良く吟味してから付けたのである。だから、私は、「BSD 4.2 (4.3) は、こんな Primitiveが付いたから、こんなことが記述できる様にな、たぞ。偉いだろう。」と、UNIX大好き少年が、言うのを

4とおと、「一週間待っている、俺が本社のOSを見せやる」と言いかけてしまう。(実際には、初場で「そんな物は、OSに最初から絶対必要なもので、RSXには付いている」と言ってしまうのだった)

㊦ とせば、Process間通信が親子の間でしかできないのは、大問題である。(BSDでは「SOCKET」、SystemVでは「Named Pipe」で解決されたが、そんなもの、最初から大問題だし、非線形マジの難しさが残る。)

㊧ も、UNIXを、みんなが世界中で使っているのは、重要なし、もっと重要なのは、みんなNetworkでつながっていることだ。

㊨ UNETというのがある、これは中心にあるSignalは一つもなく、各自バラバラのUNIXマシンが電話線でつながっているだけなのだが、これがすぐれもの、mailと「エ、ユー、リ、テ、エ」手紙を送ると、それがすぐ近所のUNIXマシンへ行き、そのマシンからまた次のマシンへ、「バケツリ」の要領で運ばれて、目的の人にまで届く。それが他人様のマシン(と電話)を

借財ので、自分がお金を払うのは、最初の¥10 たけで、あとは
 ティム(手紙)が自分で、目的地までさがして、みんかの力を借りて
 相手に届く。そして JUNETは、DARPAにも、乗り入れているので、
 アリカにも、届く(たぶん、思)。東京には、kosaka@astecと
 いう人が、手紙を出して、返事ももらった。

News も、おもしろい。基本的には、Networkと、同じで、Newsが
 電話から、勝手に Loginして来て、勝手に Fileをいれて来る。そして、
 自分のマシンの時分が来ると、勝手に電話をかけて、他人のマシンの
 Newsを送る。なかなか気色かええ。

この Newsは、役立つものもたくさんあるし、UNIXのユーティリティは
 もちろん MS-DOSやCP/Mや、Amigaや Macintoshのゲームも
 送られていて、便利!!

その他、最近の研究者の手紙のやりとりは、ほとんど"mail"で
 行われていて、KITのおお先生に「電話を、なんとしても連絡が
 つかないから、...」と言われた京大の「H君」(H君の手紙の筆者)は
 「mailで出してくれば、良いのに...」と、一人ごとを、言っておられた。

④ たくしは、PDP-11を持っていると言ってきたが、今は、VT-125 Emulatorで、Sunに loginしているらしい毎日。Professional-350は、keyとCRTはVT-220みたいなので、カーソルは良い。VT Emulatorも、Newsで流れて来た「VT test」なる Programで、ほぼ満点だったの、調子は、仲々良い。でも、RSXはやっぱり UNIXの、はるかにしかりしている。

⑤ Window Systemは、Domainのが良いね。と言って、Domainが Pro-350のと割りに置いてあるのに、よー使わね。

⑥ Editorは Emacs (又は GNU Emacs) と、言っ人が居るか、viの方が、楽だね。でも、本当に欲しいのは Word Star (又は Word Master) だよ。

⑦ このがれの Window Systemで、(速さは多少目をつむれば) 使いものになるのは、Macintoshだけだね、Sunは、たぶん X-windowも、Cut & Pasteは、やりにくい。Domainは、Moreを動かすのは、Macよりも良い。それで、私は、VT-125を使っているのだよ。かきひ。

⑧ ねえ、ucrcの切り口を、UNIXのコンパイルで、中身の作りは OS を、つくってくれ。 (TRONは、ちゃんとやめてほしいな)

(18/Nov/1986)

BSD v.s. System V

通産省に \sum 計画と、いのがあって、それのO.S.は
 絶対的に UNIX System V と決まっている。しかし、System V は、
 BSD (パークレイズ、改造された UNIX) より、進化が遅かった分を、
 AT&T (Bell Lab.) の本家の意地で、急速に、しかも BSD と違う形で、
 改良している。おかげで、BSD と System V は、大分ちがうものになっ
 (また、また、そのやり方をぼんやり (私は詳しくないの、と云っても、ぼんやり
 にはある) 見ていて、System V の方が、官僚的でお臭いがる。
 その上に日本では通産省であるから、もう System V は、がんばらぬ。

② へ A SONY の NEWS である。BSD であるから、もう日本では
 BSD は、「自由」「文化」ときて、「研究者」「スエカザン有奴」と。
 いうことで、Hacker や Wizard は、BSD が欲しくて、BSD が
 載るのは、そのおかげで SONY の NEWS のみ。

③ かも NEWS の NFS や、Network Disk は Sun Microsystems の
 製品だから、多分、SONY と同じ。

④、心がある人々は、みんな BSD が欲しくて、NEWS を買っ
 (VAX-Station II も、少し売れる)

⑤ は、SONY が好きでもない、UNIX は、みんな好きだ。

コンピュータで飯を喰べたい方へ。

- ①の(註ない)文章を讀む人の中で、本当に計算機の先進的な分野で飯を喰いたいと思っている人がいれば、以下を参考にして下さい。
- ② 近にあるものは、使い込む様に、難し過ぎる物は、突っ込み、得意なものを骨までしゃぶる様に。
- ③ プログラムなど、くさるほど書いて当然なので、「C」の様な言語でなら、文字列のscanや、String操作が、すぐできて、できる。BASICの様な言語のインタープリタや、コンパイラが書いたことがあるとか、グラフィックのパッケージを作ったことがあるとか、とにかく、完結した、大きさのある物を、学部生のうちに(できることなら2回生くらいに)書く様に。
- ④ P/MやMS-DOSなどのコンパ外もプログラムのふるまひは十分に知ると、どの様に実現におおきき考えおく。または、中身を讀んでほめて勉強にも良い。TVゲームを讀むのと、CP/Mを讀むのは、大差ないが、CP/Mは、世の中で言話が通ずる、かがあるらCP/MがUNIXなら、もっと良い。

(本) 論をやる。O.S. ごと、日本では、特別な本はないが、本を讀む。先端から、論理型とか、Object Orientedとか、Actorとか、並列処理とか、人工知能とか、知言語工学とか、心理学とか、神経学とか、何人という、なれてもやっておくのが、足腰の基礎体力になる。(UNIXの構造の本は、あまり勉強にならないが、やるといいよ)

(ハ) ハードウェアもやる。ハードウェアが得意な奴も、先に言った様にプログラムは、くさる程書けて、あたり前。ソフトウェアが得意な奴も、自分のマシンがど人も風にならなくて、どういうO.S.が、どういうふるまひをしているかを、考えられて、あたり前。大体、ふるまひを知らずに、どうやって速いソフトウェアを作ることが出来るか! ハードウェア、全部とは言わないが、手近にあるマシンのCPUが、どの様に動いていて、I/O 操盤が、どう動くか動いているかを、しゃぶる程知れば、他のマシンも大体わかる。

(ア) プログラムの修業と、ハードウェアの修業は、学部生のうちに、片付けておく。

(上) 記のことなど、「お前は無理解な奴」と言われるかも知れないが、私の今居る所には、せいぜいマシンのことを詳しく知っているから、O.S. に詳しくなったり、Temporal Logic を研究したりしている奴が、ゴロゴロ居る訳で、私に、できれば、少なくとも上記のことをこなしている人とは、いっしょにお仕事したくはないなあ。

編集後記

学祭もすでに2日を過ぎましたが、
今だに、完動しているのは入室チャイム
だけであり、しかもこの入室チャイムも、
2日目の午後、突然の福井学長の見
学通告により、あわてて基板を作り大
橋さんや並河さんの多大なる助力
のもとに、88に「いらっやいませ」を
しゃべらせようとセッティングを行なっ
たが、とうとう、学長が訪れる時間ま
でに間に合わず、大恥をかいてし
てしまった。あと1時間みれば完動
していたのだ……。

現在は、小前さんと駒嵐さんがマイ
クロマウスの製作に没頭していま
すが、最終日までに動くのでし
ょうか……。

1986.11.23 AM.2:00頃

物部 隆之



©

COPYRIGHT BY K.I.T. Computer Club 1986.11