

# I 部 ノウノウ編

## 第 I 部をお読みになる前に

第 I 部では、X1 シリーズの最新マシンである X1turboZ について、その特徴であるグラフィックス機能を中心に解説してゆきます。各章の内容は次のとおりです。

第 1 章 コンピュータグラフィックスを行う場合に最低限必要とされる事柄について解説します。

第 2 章 X1turboZ に特徴的なグラフィックス機能を、BASIC でグラフィックツールをすることにより解説します。

第 3 章 X1 シリーズのオプションとして発売された立体映像セット (液晶シャッター方式の 3 D メガネ) を含めたステレオグラフィックスについての理論を解説します。

第 4 章 FM 音源に関する基本的な事柄について解説します。

第 I 部は基本的には、X1turboZ ユーザーを対象に書かれています。従って、第 I 部で掲載したプログラムは turboZ 以外の機種に関して、動作確認を行っていませんので、turboZ 以外の機種をお持ちの方はご注意ください。

# 第1章

## グラフィックスの理論

コンピュータグラフィックス(Computer Graphics 略してCGと呼ぶ)というと読者の皆さんはどのようなものを思い浮かべるでしょうか。多分、テレビのコマーシャルや映画等で使用されているきらびやかな映像をイメージされることと思われます。

しかし、そのようなきらびやかな映像も、元を問いただせば頭にコンピュータとつく以上、当たり前ですがコンピュータによって作られたものなのです。従ってCG理論の基本というのはコンピュータ上でどのようにしてグラフィックス処理を行うかという、これもまた至極当り前の所から始まっているといえます。

本章では、コンピュータとしてX1turboZを使用した場合のコンピュータグラフィックスに関して、最低限知っておかなければならない事柄を解説します。

### 1-1 I/Oポート

パソコンは、CPUだけですべての処理を行うわけではなく、SIO,CTCなどという周辺LSIにもある程度処理を任せて、処理の高速化を計っています。I/Oポートは、これらの周辺LSIとやりとりするための窓口のようなものです。パソコンを一つの都市として考えると、CPUは中央指令部で、メモリは住所(アドレス)を持った町と考えられます。そうするとI/Oポートは、港のようなものでしょうか。

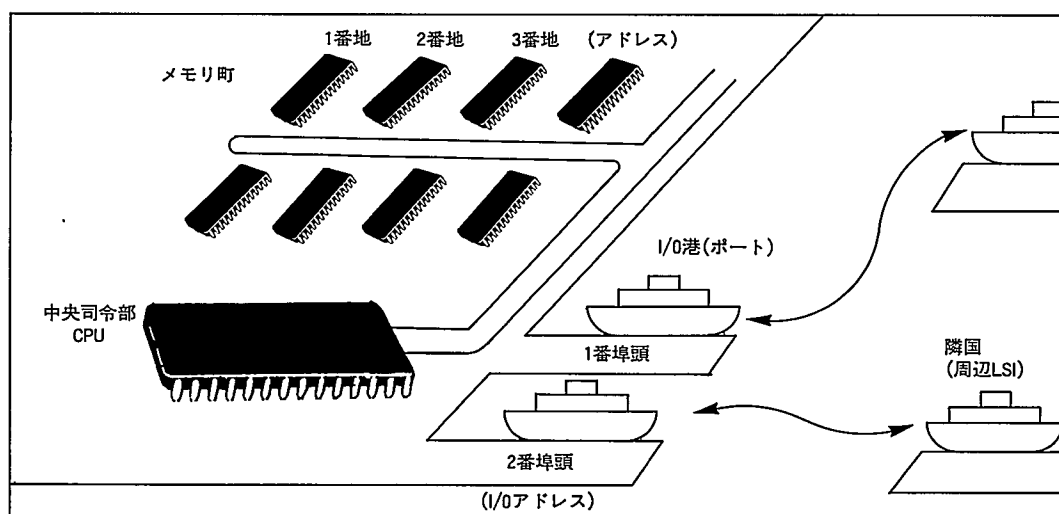


図1-1 I/Oポートは周辺装置とのデータを入出力する港のこと

コンピュータ用語では、アドレス、ポート、マップなど都市にたとえられた表現がよく使われます。これは基板上に立ち並ぶIC群があたかも都市のように見えるためなのかもしれません。

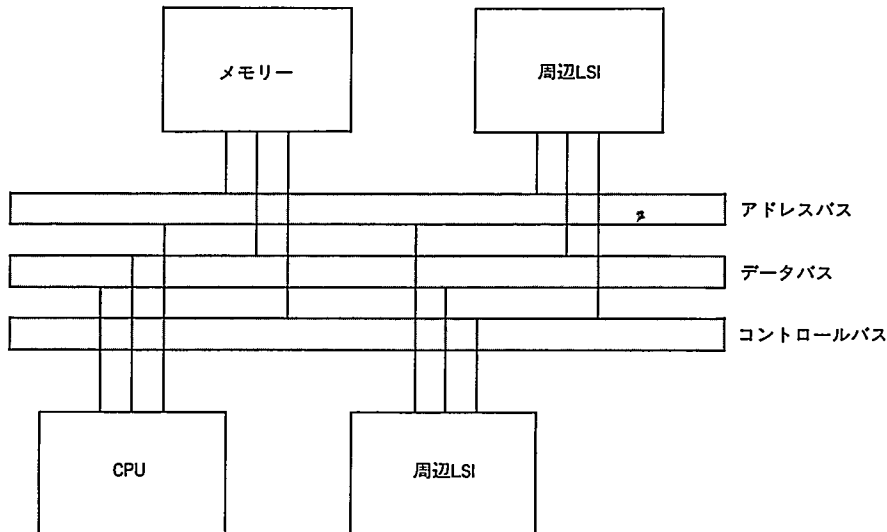


図1-2 パーソナルコンピュータのブロック図(図1-1と同じ意味)

BASICのOUT &H1FB0,&H80というコマンドは&H1FB0で指定される周辺LSIに対して、&H80というコマンド又はデータを送るということです。

OUT    &H1FB0,    &H80  
出力する   どこへ   何を  
(何番埠頭へ)

逆に A=INP(&H1FB0)は、&H1FB0 で指定される周辺 LSI から、データを読み込み、その読み込んだデータを Aに入れるということです。

A = INP (&H1FB0)  
 どこへ 入力する どこから  
 (何番地へ) (何番埠頭から)

指定された周辺 LSI に対して、データ又はコマンドを送るといって、指定された住所をたよりに郵便屋さんが手紙を届けるイメージに受けとられます。このように考えても全く問題はないのですが、実際に CPU が内部で行っている処理は多少異なっています。CPU は、周辺 LSI に対して、それぞれ別々のコマンドを送っているわけではなく、すべての周辺 LSI に対して同じコマンドを送っているわけです。それを周辺 LSI 側で、自分に対して送られたコマンドかどうか判断し（これをデコードするといいます）自分に対して送られたコマンドのみを実行します。

従って内部処理的にみると、CPU は全ての周辺 LSI にスピーカーで呼びかけていると考えた方が良いでしょう。

先程の例だと、

「&H1FB0 さん、&H80 のコマンドを実行して下さい」

と呼びかけているわけです。

よく I/O マップなどに、&H0F \*\* という記述がなされていることがありますが、これは下位 2 バイトはなんでもよいということを意味しています。つまり、I/O ポートを呼び出すとき、フルネームで呼び出さず、名字だけで呼び出すようなものでしょう（一般にこのような場合アコードしていないといいます）。

## 1-2 V-RAM 構成

?

画面上に表示されるデータが保存されているメモリーのことを V-RAM と呼びます。V-RAM には、キャラクタコードを保存しているテキスト V-RAM と画面の 1 ドット毎の色情報を保存しているグラフィック V-RAM とがあります。このグラフィック V-RAM も色情報のデータの持ち方によって、パレットモード（コンパチモード）と多色モードに分かれます。この多色モードは、X1turboZ によって拡張されたモードです。

V-RAMの構成

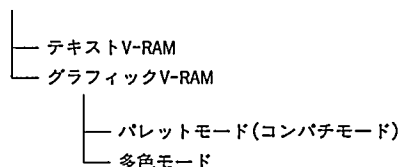


図1-3

※パレットモードのことをマニュアルでは、コンパチモードと呼んでいます。しかし、コンパチモードとここで表記すると、更に新しい機種が出たときに、その機種の多色モードもコンパチモードと呼ばないわけにはいなくなってしまう混乱する可能性があります。そのため、コンパチモードをパレットモードと表記することにします。

### ●パレットモードと多色モードについて

パレットモードとは、各画素の色コードを V-RAM 上に持つモードです。色コードとは、パレットレジスタ（ルックアップテーブルともいう）のインデックス値のことであり、X1turboZ では、このインデックスを、アドレスレジスタとデータレジスタに分けて設定します。画面上には、このパレットレジスタにより、R,G,B の各輝度に変換されて表示されます。

これに対して多色モードとは、各画素の輝度を V-RAM 上に持つモードです。即ち、各ピクセル毎に R,G,B それぞれの輝度を直接持つわけです。パレットモードでは色数が V-RAM の枚数により制限されてしまいますが、多色モードの場合このような制限は全くありません。

このように V-RAM の使い方には 2 通りあり、各々特徴的な機能を持っていますが、パレットモードに対して多色モードでは、パレットレジスタを固定的に使用しているだけなので、ハードウェアは、全く共通です。

I 部の最後に、4096色同時表示モードと、64色 2 画面モードのパレットレジスタ初期化プログラムを掲載します。4096色モードから64色 2 画面モードへと切り変えるにはこのパレットレジスタ初期化プログラムを起動させないと正常な画面は表示されません。

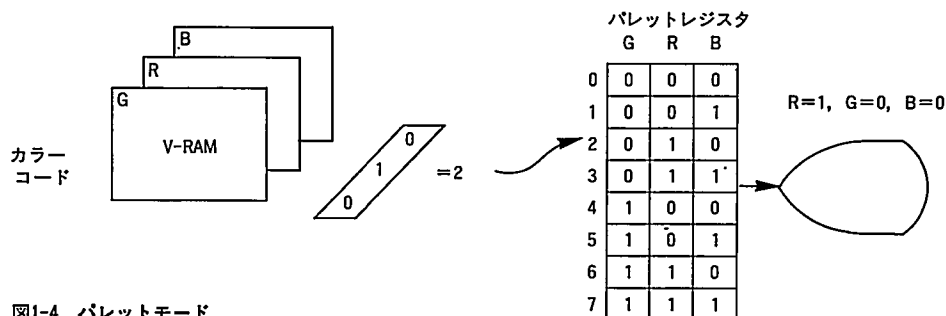


図1-4 パレットモード

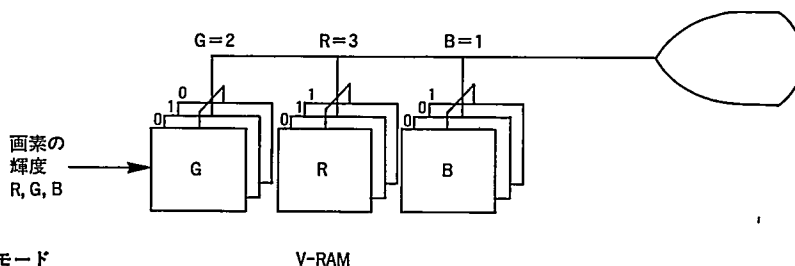


図1-5 多色モード

V-RAM

## ●パレットモードと多色モードの違い

	パレットモード	多色モード
色数	V-RAMの数により制限あり。 例：3面のときは8色	制限なし
階調数	階調数を増やすと色数が減るため、 階調表現がしにくい。	V-RAMの数 12面のとき RGB 各 4 階調 6面のとき RGB 各 2 階調
パレット機能	有	無
使用法	色コード毎に色をリアルタイムに変更 できるため、色の配色を見るとき に便利。	R, G, Bの輝度を直接持つため、ビ デオ画像の取り込みや、レイトレー シングなどのコンピュータグラフィ ックスに便利

表1-1 パレットモードと多色モードの違い

## 1-3 V-RAM のアドレス

表示モードは解像度によって数種類あります。しかし、大きく分類すると、縦方向と横方向の解像度により、次表のように分類できます。V-RAM のアドレスは、このモードにより決定されます。

		横方向の解像度	
		WIDTH 40 320ドット	WIDTH 80 640ドット
縦方向の 解像度	200ドット	320×200	640×200
	400ドット 192(384)	320×400 320×192(384)	640×400 640×192(384)

表1-2 ディスプレイの解像度 (X1turboZ)

192ドットモードは、低解像度モニターで高解像度モニターでしか使えない400ドットモードの漢字等を表示するために設けられた互換モードです。従って V-RAM のアドレスは400ドットモードとほぼ同じ構成となっています。

各モードにおける画面左上の部分の V-RAM アドレスを比較したものを次に示します。

I 部の最後に、VRAM のアドレス計算プログラムを掲載してあります。

横 縦	320 200	640 200	320 400 (192/84)	640 400 (192/384)
1	4000	4001	4000	4001
2	4800	4801	4400	4401
3	5000	5001	4800	4801
4	5800	5801	4C00	4C01
5	6000	6001	5000	5001
6	6800	6801	5400	5401
7	7000	7001	5800	5801
8	7800	7801	5C00	5C01
1	4028	4029	6000	6001
2	4828	4829	6400	6401
3	5028	5029	6800	6801
4	5828	5829	6C00	6C01
5	6028	6029	7000	7001
6	6828	6829	7400	7401
7	7028	7029	7800	7801
8	7828	7829	7C00	7C01
			4028	4029
			4428	4429
			4828	4829
			4C28	4C29
			5028	5029
			5428	5429
			5828	5829
			5C28	5C29
			6028	6029
			6428	6429
			6828	6829
			6C28	6C29
			7028	7029
			7428	7429
			7828	7829
			7C28	7C29
			4050	4051
			4450	4451
			4850	4851
			4C50	4C51
			5050	5051
			5450	5451
			5850	5851
			5C50	5C51
			6050	6051
			6450	6451
			6850	6851
			6C50	6C51
			7050	7051
			7450	7451
			7850	7851
			7C50	7C51

表1-3 画面左上のV-RAMアドレスの比較

● V-RAM のアドレスの求め方

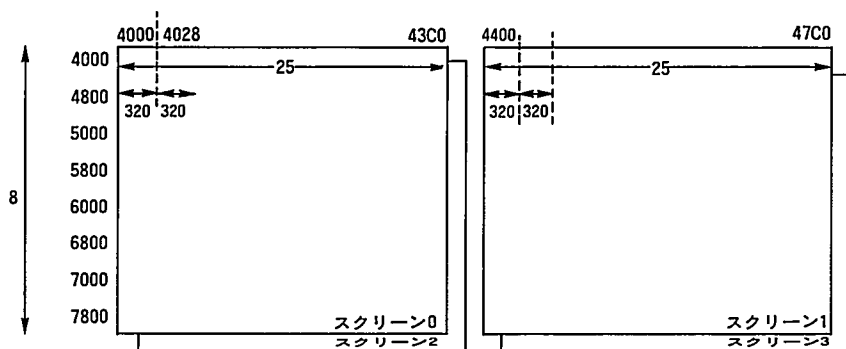


図1-6 320×200ドット(4画面)

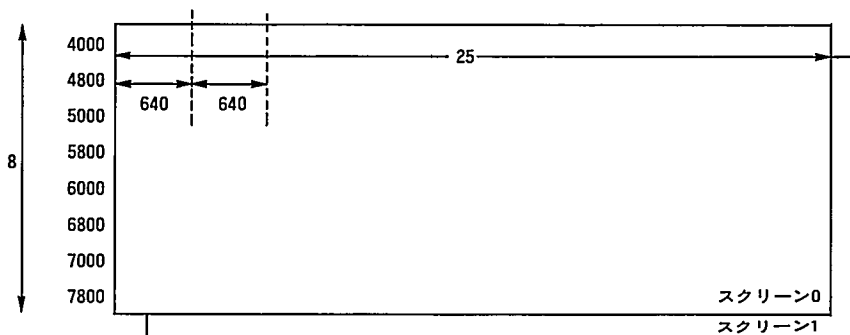


図1-7 640×200ドット(2画面)

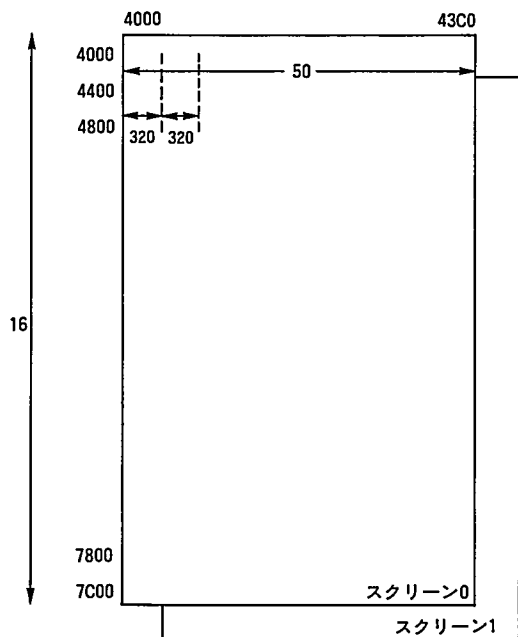


図1-8 320×400(2画面)



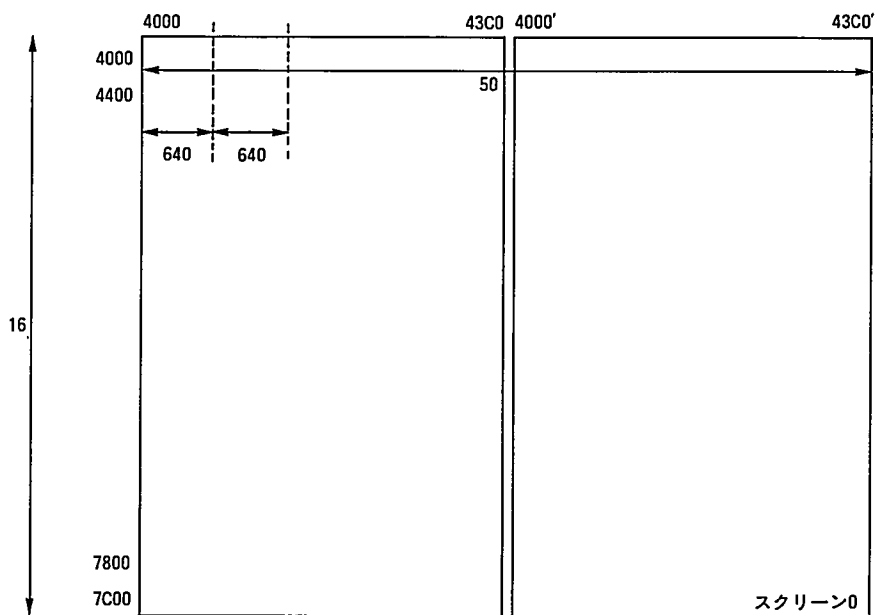


図1-9 640×400(1画面)

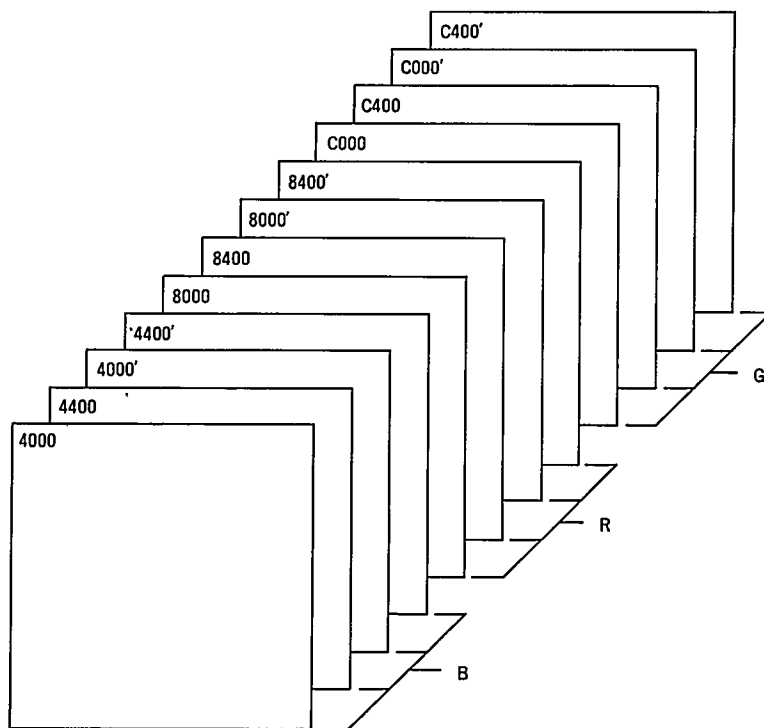


図1-10 320×200(4096色モード)

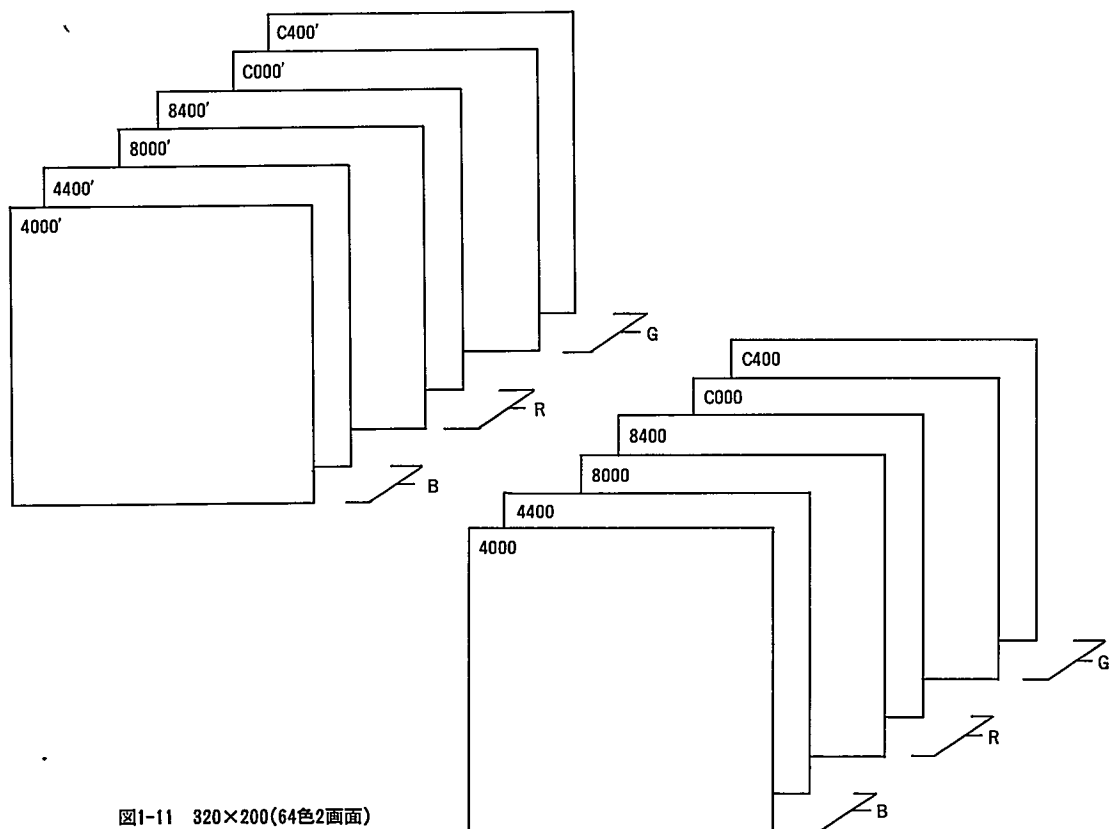


図1-11 320×200(64色2画面)

## 1-4 ディスプレイの表示の仕組み

ここでテレビ信号の仕様について説明します。

テレビ画面は、RGBの蛍光体に光を当てることによって表示しています。しかし、ある瞬間を見てみると画面上の1ドットしか光っていません。その動作をスローモーションで見るとよく分かるのですが、結局、画面の左上から順番に1ドットずつ光らせているというわけです。画面がちらつきもなく自然に見えるのは、蛍光体がしばらくの間光り続けているのと人間の目の残像現象によるためです。

また、NTSC信号の場合、走査速度が遅いため、525本の走査線を1本おきに表示するインターレース方式を採用しています。これに対して、高解像度モードでは、1本ずつ表示しています。このような方式を、ノンインターレースといいます。ノンインターレースは、インターレースに比べてちらつきが少なく、見やすい画面となります。

参考までに、EDTVではノンインターレース方式にすることにより、現在のTV放送の画質の向上を目指しています。

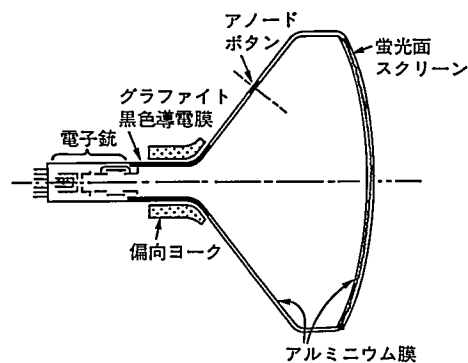
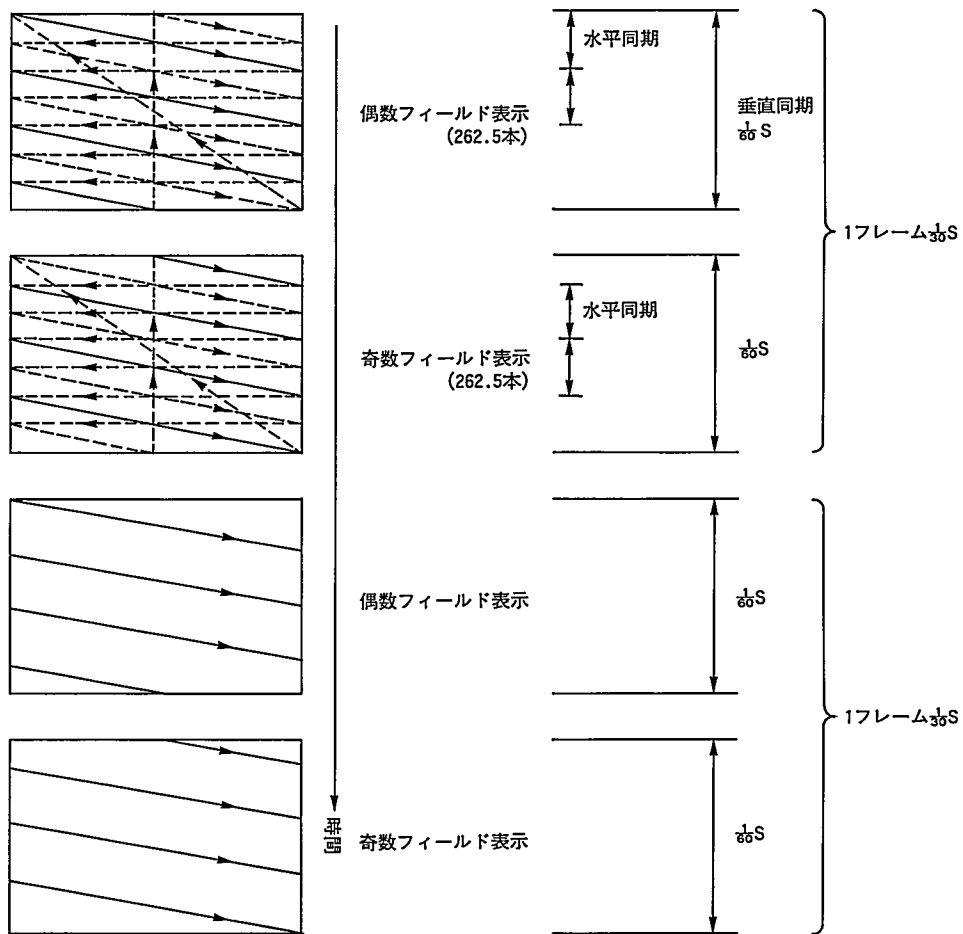


図1-12

テレビの1画面は、偶数フィールドと奇数フィールドから成り立っています。このように、1回の垂直同期の間に表示される262.5本の画像を1フィールドと呼びます。また、偶数フィールドと奇数フィールドを併せたものを、1フレームと呼びます。

1フィールドを走査するのに1/60秒かかるため1フレームの走査では1/30秒かかります。従って、1秒間に30枚の画面を見ていることになります。

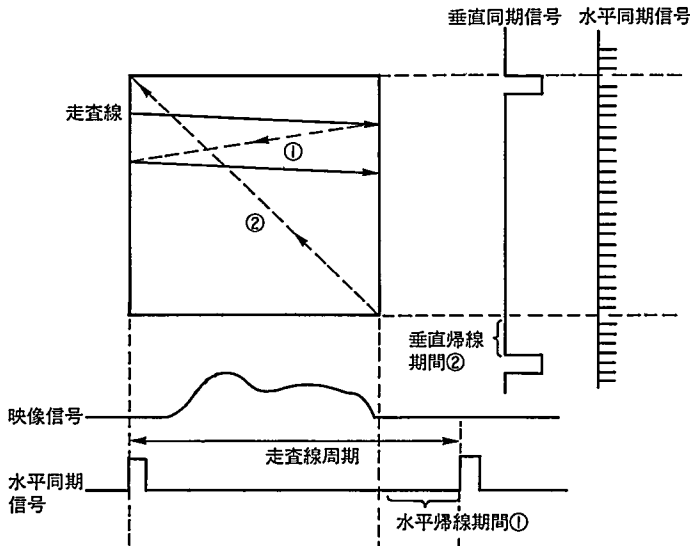


図1-13

## 1-5 ウィンドウと座標変換

ディスプレイの表示解像度は、使用しているパソコン、及びディスプレイの機種によって異なるのは当然ですが、同一パソコンで複数の表示モードをサポートしているものもあります。この場合、使用するソフトウェアが必要とする色数、解像度に表示を合わせて使用することができます。日本語ワードプロセッサや、表計算ソフトウェアのように色数が少なくても解像度が高いことを要求されるソフトウェアにはそのように合わせ、グラフィックツールやレイトレーシングを行うソフトウェアのように解像度は少し落ちて色数が多いことが要求されるソフトウェアにはそのように合わせるという具合に、タイプによって解像度を切り換えるという使い方ができます。

XlturboZでも、

640×400ドット	320×400ドット
640×384ドット	320×384ドット
640×200ドット	320×200ドット
640×192ドット	320×192ドット

の各表示モードがサポートされています。

この表示モードのように画面の1ドットに対応する座標系を、画面座標系又は、デバイス座標系と呼びます。この座標系は、物理的に決定されます。

しかし、アプリケーションソフトウェアを作成するとき、このような物理的な座標系を意識して作ると、汎用性がなくなり、移植性も悪くなってしまいます。

そこで、ユーザー座標系と呼ばれる論理的な座標系を定義できるようになっています。ユーザー座標系は、ワールド座標系ともよばれる実数の範囲で定義される広い空間です。このユーザー座標系と画面座標系を対応づけることをウィンドウ、ビューポート変換と呼びます。これは BASIC の WINDOW コマンドに対応します。

$$\text{WINDOW } \underbrace{(X_s, Y_s) - (X_e, Y_e)}_{\text{この値をビューポートと呼ぶ}} , \underbrace{(X_1, Y_1) - (X_2, Y_2)}_{\text{この値をウィンドウと呼ぶ}}$$

X1 の BASIC においては、WINDOW という 1 つのコマンドとしてインプリメントされていますが、一般的には、ビューポートという概念とウィンドウという概念に分かれますので、ここでは分けて説明します。

ウィンドウとは、ユーザー座標系のどの部分を表示するかを示すものです。言葉の通り、ユーザー座標系の中に窓をあけて、その中だけが表示されるように決めるわけです。

これに対して、ビューポートとは、ウィンドウによって決められた範囲を画面上のどの部分に表示するかを決めるわけです。

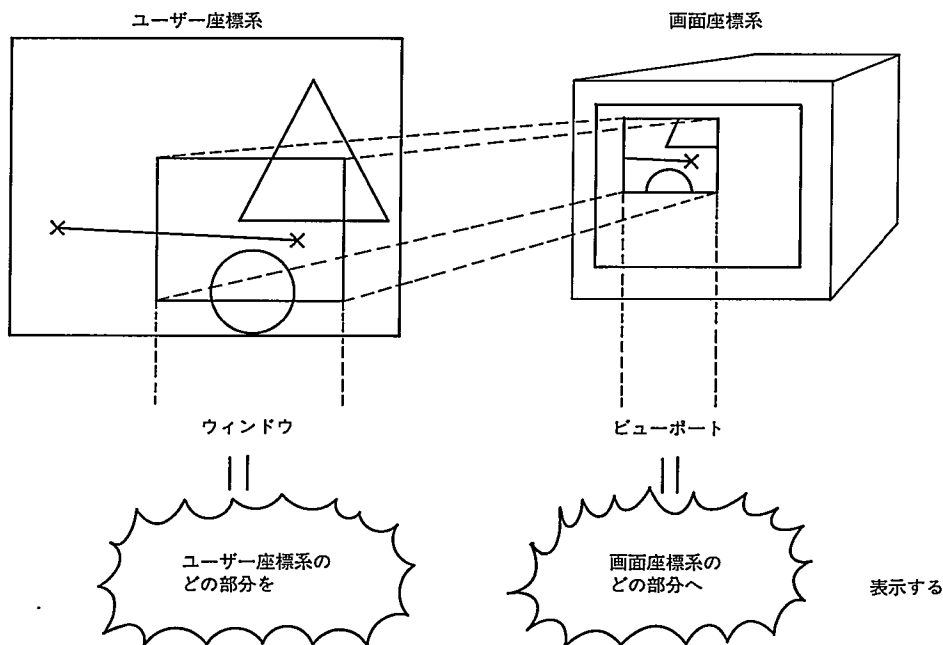


図1-14 ユーザー座標系と画面座標系の対応

これらの、ウィンドウや、ビューポートにより、ユーザー座標系で与えられたデータが、どのように表示されるかを表す変換マトリクスが生成されます。このマトリクスは $3 \times 3$ のマトリクスで表され平行移動とスケーリングが行われます。

ここで、ウィンドウ、ビューポート変換マトリクスについて説明します。

ビューポート (Xs, Ys) – (Xe, Ye)

ウィンドウ (X1, Y1) – (X2, Y2)

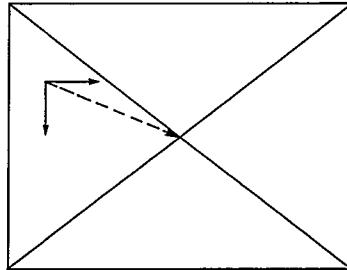


図1-15 平行移動(ウィンドウの中心へ移動する)

まず、ユーザー座標系の原点を、ウィンドウの中心へ移動する変換マトリクスを求めます。

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{X_1+X_2}{2} & -\frac{Y_1+Y_2}{2} & 1 \end{bmatrix}$$

次に、ウィンドウとビューポートの大きさを合わせます。

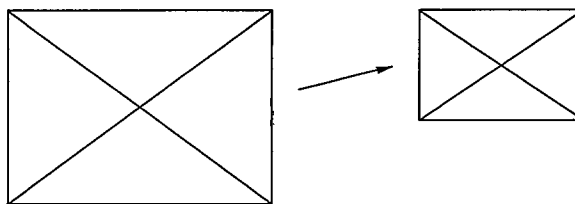


図1-16 スケール(大きさを変える)

$$M_2 = \begin{bmatrix} \frac{X_e - X_s}{X_2 - X_1} & 0 & 0 \\ 0 & \frac{Y_e - Y_s}{Y_2 - Y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

最後にビューポートを画面上に配置する変換マトリクスを求めます。

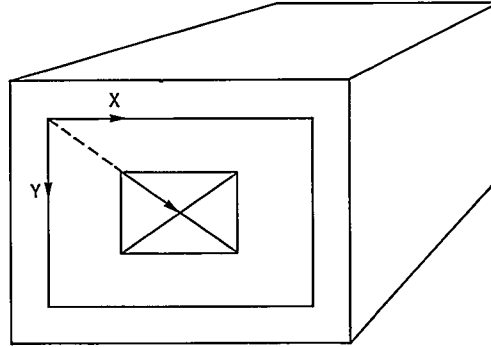


図1-17 平行移動(画面上の表示位置へ移動する)

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{X_s + X_o}{2} & \frac{Y_s + Y_o}{2} & 1 \end{bmatrix}$$

従って、点(X, Y)を変換するとき M1, M2, M3 を掛けあわせればよいため、変換後の点(X', Y') は

$$(X', Y', 1) = (X, Y, 1) \cdot M_1 \cdot M_2 \cdot M_3$$

となります。

この変換のプログラムをI部の最後に掲載してあります。

## 1-6 色の表現(RGB と HSV)

### (1) 色の表現方法

色の表現方法にはいろいろありますが、代表的なものとして以下の方法があります。

- ① RGB モデル パーソナルコンピュータ, グラフィックディスプレイ
- ② HSV モデル 美術, デザイン
- ③ CMY モデル 印刷, 写真のネガ

パーソナルコンピュータは, RGB モデルであり V-RAM や, パレットレジスタの設定は, RGB で行われます。しかし, 美術や, デザインの世界では使いにくいため, HSV 方式がよく使われます。CMY モデルは, 特殊な用途で使われるため, 説明は省略します。

### (2) RGB モデルについて

RGB モデルは, 直交座標系で表され, Red, Green, Blue の 3 色の明るさを x, y, z 軸とします。

以下に図で示します。

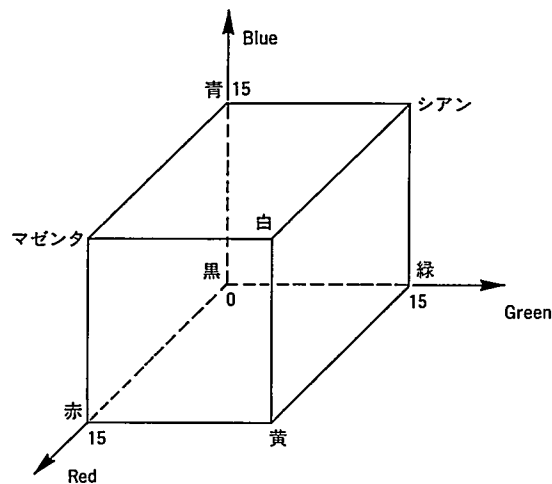


図1-18 RGBモデル

X1turboZ は, R,G,B,各 4 ビットの階調を持つので  $2^{4 \times 3} = 4096$  色を表現することができます。

### (3) HSV モデル

HSV モデルは, 色相(HUE), 彩度(SATURATION), 明度(VALUE)の 3 つの要素で表されます。

色相とは, 色あいとも呼ばれるもので, 赤青黄などの色の種類を表します。これは, 光のスペクトルにより変化します。ただし, 波長によるスペクトルではなく仮想的なスペクトルで, 色度図と呼ばれます。

彩度とは, 飽和度, あざやかさと呼ばれるもので色の濃淡(純粋さ)を表します。これは, 無彩色(白, 灰, 黒)の割合により変化します。

明度とは, 輝度, コントラストと呼ばれるもので, 色の明るさを表します。これは, 光の強さにより変化します。

これらの 3 つの要素を, 色立体で表すと図1-19のようになります。中心軸の下から上にいくにつれて明度が増加し, また, この軸を中心にして回転する円周上に色相が表され, 角度として表されます。またこのときの半径が彩度を表し, 半径が大きいほど純度が増します。



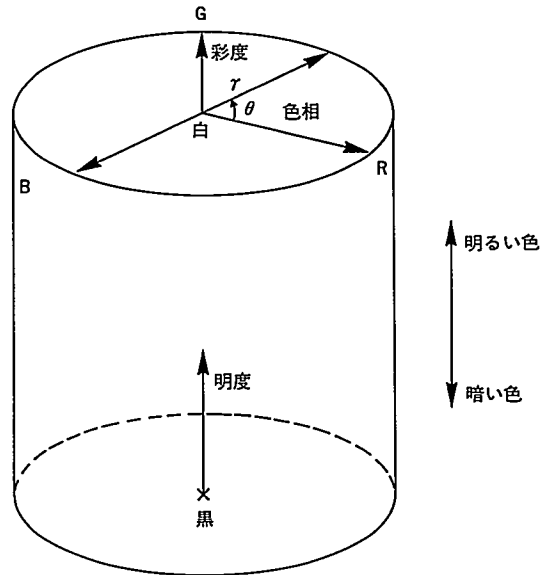


図1-19 HSVモデル

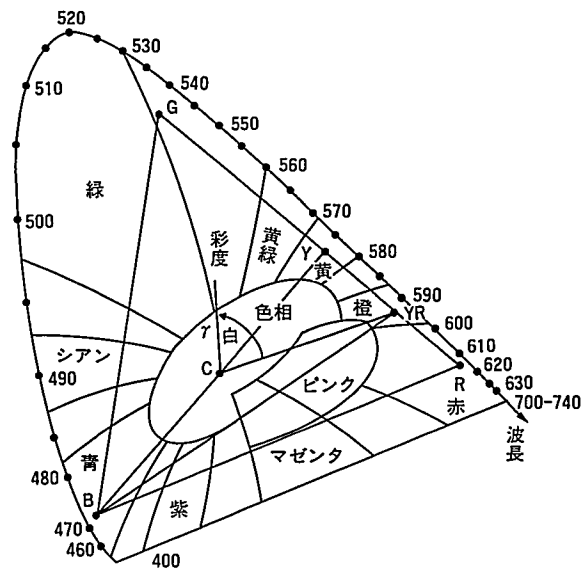


図1-20 色度図

各色の要素とスペクトルと輝度レベルの関係を図で示します。

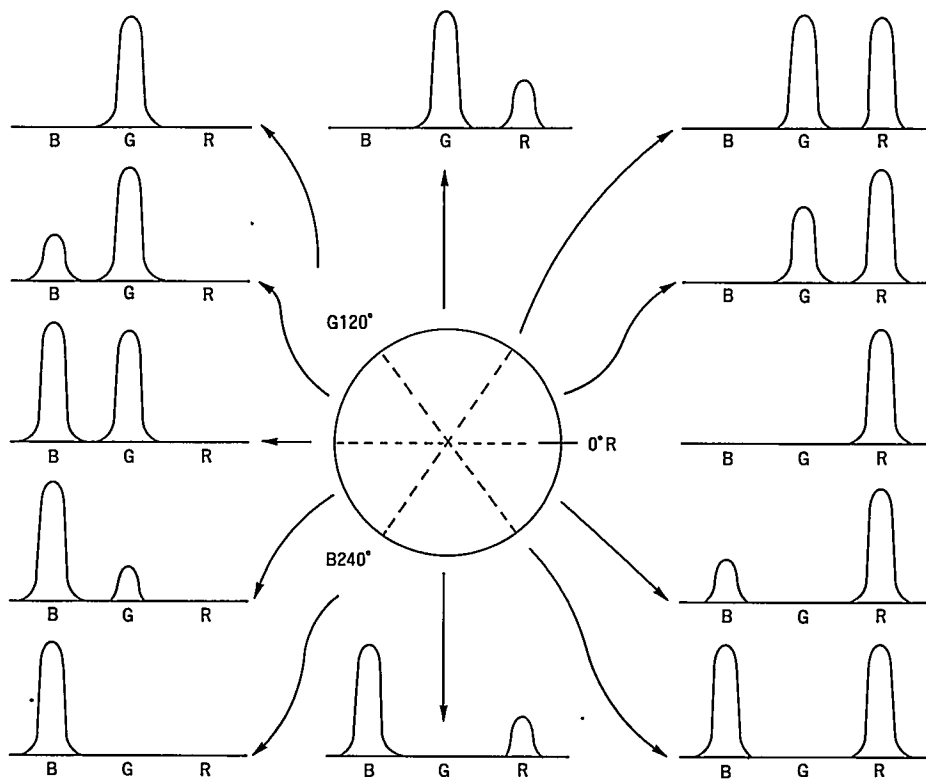
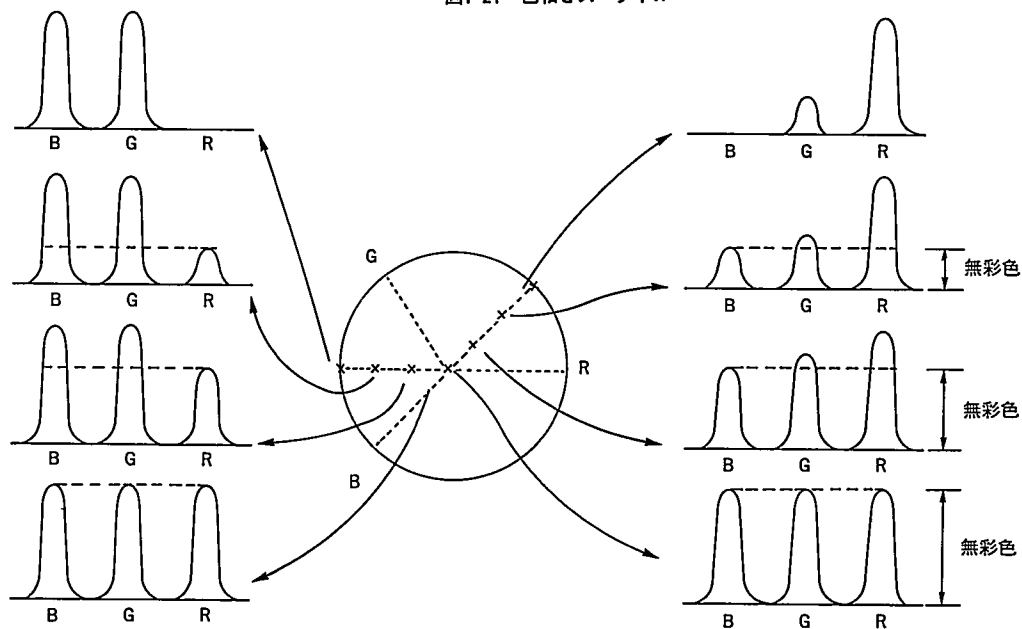


図1-21 色相とスペクトル



R, G, Bが合成されると、白色となり無彩色となる。  
彩度とは、無彩色でない部分の比率である。

図1-22 彩度とスペクトル

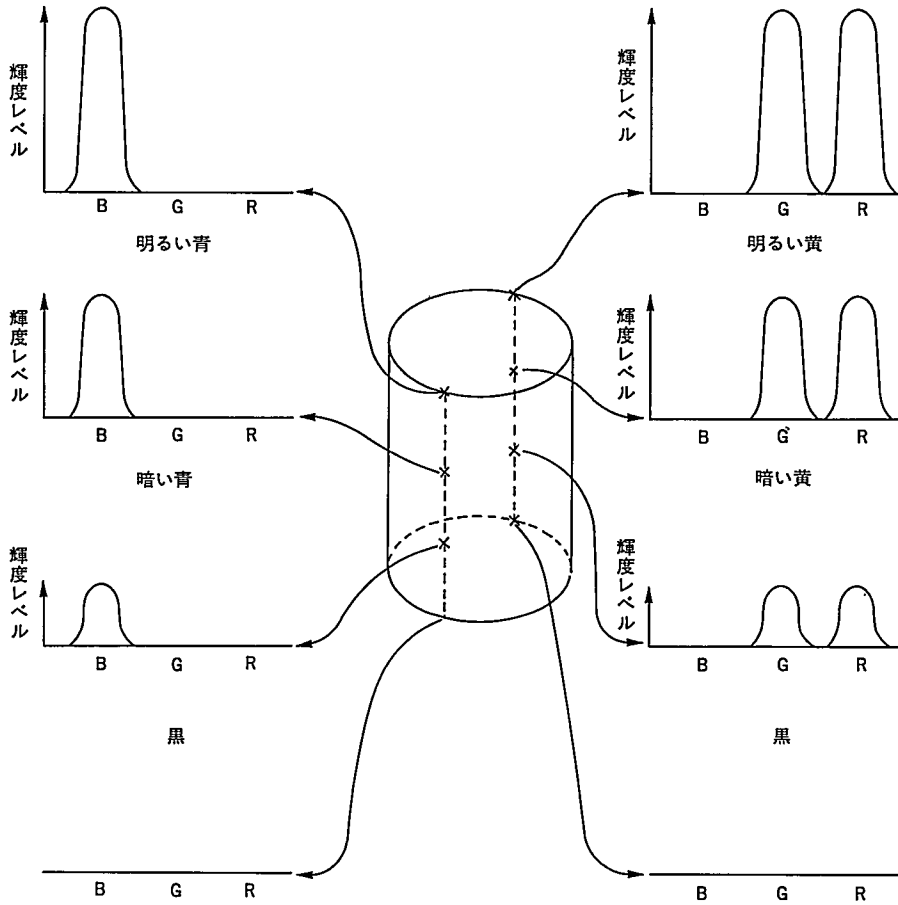


図1-23 明度とスペクトル

(4) HSV モデルから RGB モデルへの変換と逆変換

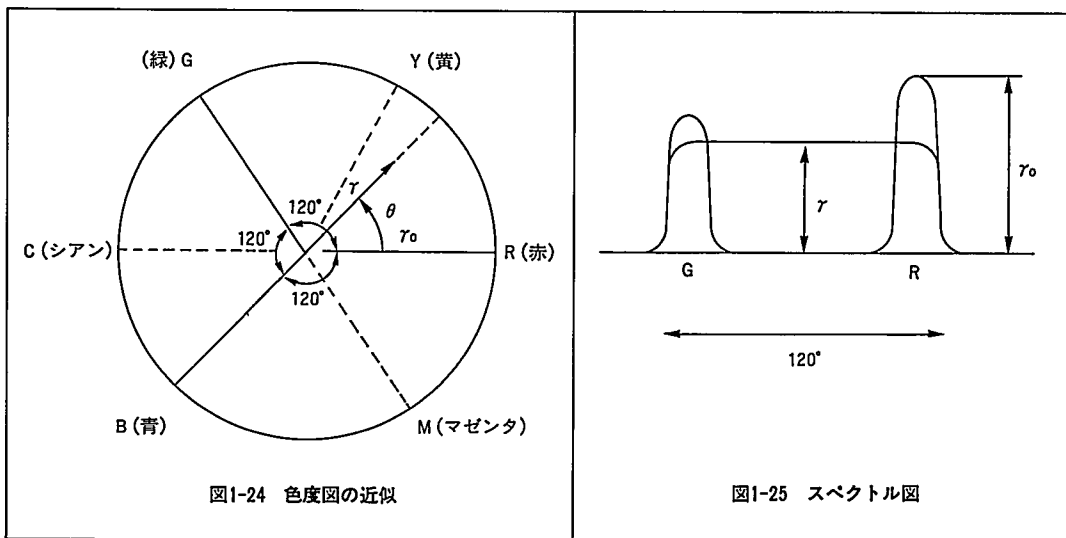


図1-24 色度図の近似

図1-25 スペクトル図

図1-20の色度図を円柱で近似すると図1-24のようになります。

赤を $0^\circ$ とすると、緑は $120^\circ$ 、青は $240^\circ$ となります。色相は、度で表す方法とラジアンで表す方法がありますが、度の方が分かりやすいので、度で表すことにします。また彩度と明度は $0 \sim 100\%$ で表すこととします。

RGB、それぞれの階調数を $n$ 、色相を $H$ とすると

$$0^\circ \leq H \leq 60^\circ \quad R_1 = n$$

$$60^\circ \leq H \leq 180^\circ \quad G_1 = n$$

$$180^\circ \leq H \leq 300^\circ \quad B_1 = n$$

$$300^\circ \leq H \leq 360^\circ \quad R_1 = n$$

となります。ここで、彩度 $S$ は $1 - \frac{r}{r_0}$ となるから

$$0^\circ \leq H \leq 120^\circ \quad B_1 = \frac{n \times (100 - S)}{100}$$

$$120^\circ \leq H \leq 240^\circ \quad R_1 = \frac{n \times (100 - S)}{100}$$

$$240^\circ \leq H \leq 360^\circ \quad G_1 = \frac{n \times (100 - S)}{100}$$

となります。残りの1色は、色相の比率により求めることができます。

$$0^\circ \leq H \leq 60^\circ \quad G_1 = n \times \frac{(100 - S)}{100} + n \times \frac{H - 0}{60} \times \frac{S}{100}$$

$$60^\circ \leq H \leq 120^\circ \quad R_1 = n \times \frac{(100 - S)}{100} + n \times \frac{120 - H}{60} \times \frac{S}{100}$$

$$120^\circ \leq H \leq 180^\circ \quad B_1 = n \times \frac{(100 - S)}{100} + n \times \frac{H - 120}{60} \times \frac{S}{100}$$

$$180^\circ \leq H \leq 240^\circ \quad G_1 = n \times \frac{(100 - S)}{100} + n \times \frac{240 - H}{60} \times \frac{S}{100}$$

$$240^\circ \leq H \leq 300^\circ \quad R_1 = n \times \frac{(100 - S)}{100} + n \times \frac{H - 240}{60} \times \frac{S}{100}$$

$$300^\circ \leq H \leq 360^\circ \quad B_1 = n \times \frac{(100 - S)}{100} + n \times \frac{360 - H}{60} \times \frac{S}{100}$$

最後に明度 $V$ を考慮すると

$$R = \frac{R_1 \times V}{100}$$

$$G = \frac{G_1 \times V}{100}$$

$$B = \frac{B_1 \times V}{100}$$

となります。

実際にはこのように場合分けをしなくても、MOD 開放などを用いてコンパクトにコーディングできます。

I 部の最後に、HSV から RGB への変換と RGB から HSV への変換プログラムを掲載します。

## (5) 光の3原色と色の3原色

光源から出た光を合成すると明るくなり白に近付いていきます。これに対して絵の具やインクをどんどん重ね合わせていくと光を吸収し黒に近付いていきます。その理由は、絵の具やインクによって見える色は、反射光を見ているからです。つまり青く見える絵の具は、青以外の色を吸収しているということです。

光のように、合成していくと明るくなるものを加算混合といいます。これに対し絵の具のように重ねるとどんどん暗くなるものを減算混合といいます。

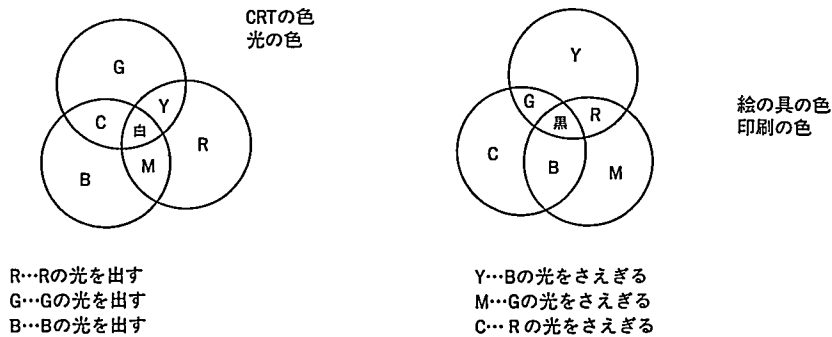


図1-26 RGBとYMCの違い

以下に加算混合，減算混合の例を示します。

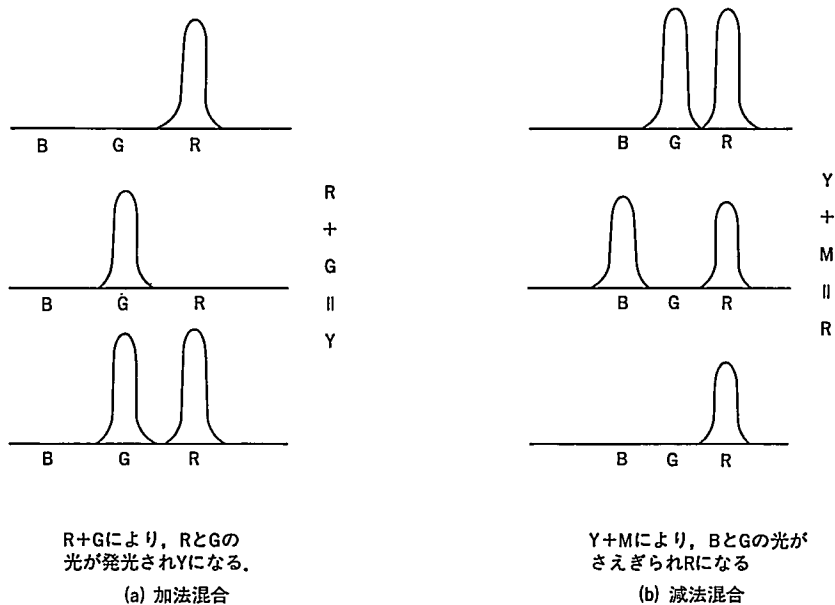


図1-27 加算混合と減法混合の違い

## 1-7 ディザリング

XlturboZでは、最高4096色の色しか表示できませんが、ディザリング機能を用いると疑似的に1670万色を同時に表示することができます。

普通、疑似的に色数を増やす方法として、タイリング方式とディザリング方式がありますが、タイリング方式は、1ドットの色の忠実性は高いのですが、解像度が落ちてしまうため、あまり実用性はありません。

ディザリング機能は、あくまでも疑似的に色数を増やすため、中間色を使用すると幾何学模様が表示されることがあります。

ディザリング方式にもいろいろありますが、本節では  $4 \times 4$  のディザマトリクスを使用する場合を例にとって説明します。

$$D = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

ここでDをディザマトリクスといいます。

表示したい点の座標値を(x, y), 表示したい色を R,G,B とします。ここで RGB は、画面座標系 (x, y) の 0-15 で表される各色の輝度レベルを表します。

i =x AND 3

J = v AND 3

上式で定義される  $i, j$  により  $D(i, j)$  を求めます.

ディザリング後の色を  $(R', G', B')$  とすると,

$$\begin{aligned} R' &= \frac{(R * 16 + D(i, j))}{16} \\ G' &= \frac{(G * 16 + D(i, j))}{16} \\ B' &= \frac{(B * 16 + D(i, j))}{16} \end{aligned}$$

となります

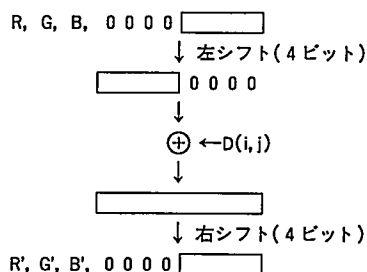


図1-28 ディザリングのしくみ

ディザ値によって輝度が高くなる確率が変わるため、色の細かい変化を表現することができません。

### ●ディザリングを使用したときの画面

VRAM が 2 枚のとき、カラーコード 0 を透明色として使用すると考えると、階調は、 $22 - 1 = 3$  階調となります。つまり、輝度が 0%，50%，100% の 3 通りしかありません。次にこれを #0%，#50%，#100% として表してディザリングを行ったときと、行なわない時の違いを示します。

#### ①ディザリングなしのとき



図1-29 ディザリングなしのときの表示

#### ②ディザリングを使用したとき



図1-30 ディザリングを使用したときの表示

遠くから見るとディザリングを使用したときは、除々に色が変化しているように見えます。