

1 グラフィックス

PDF で図形を描く方法は大きく分けて二つあります。いわゆるベクタ・グラフィックスとビットマップ画像です。T_EX ソース中に PDF オペレータを使って直接図形を描くには special コマンドを使います。

ビットマップ画像をページに表示させる方法は二つあります。

1. インライン画像
2. *Image XObject* として画像を取り込んでから **Do** オペレータによって参照する方法

XObject とは文書中のどの場所からでも何度でも参照することができる外部オブジェクト (*external Object*) です。*Image XObject* を使う方法は例えばアイコンなど複数のページに渡って何度も使用されるような画像に適しています。d_{vi}pdfm では includegraphics コマンドにより取り込まれる外部画像はすべてこの *XObject* として格納されます。

さて、インライン画像を描いてみましょう。

```
\special{pdf: content
  32 0 0 32 0 0 cm
  BI
    /Width 8
    /Height 8
    /ColorSpace /DeviceCMYK
    /BitsPerComponent 1
    /Filter [/ASCIIHexDecode]
  ID
    8040201008040201
    1080402001080402
    2010804002010804
    4020108004020108>
  EI
}
```

インライン画像は **BI** と **EI** で挟みます。**BI** と **ID** の間には画像データに関する情報が記述してあります。画像データは **ID** と **EI** に挟まれた部分にあります。**Filter** には画像データをデコードするのに必要なフィルタの種類を (フィルタをかけるべき順番に) 列挙します。この例では T_EX ソースに書き

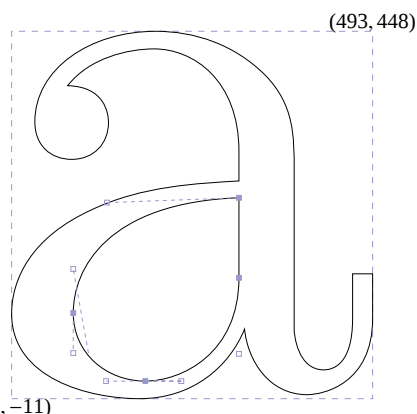


図 1: Computer Modern フォントに含まれる文字 ‘a’ のアウトライン。アウトラインは Bluesky Research より公開された PostScript Type 1 フォントから抽出した。

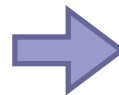
込みやすいように画像データは 16 進表記にしてある^{*1}ので ASCIIHexDecode が指定してあります。



PDF では直線や Cubic Bézier 曲線などでパスを構築して線を描いたり、パスの内部領域を塗りつぶしたりすることができます。図 1 は直線と Bézier 曲線で描いた図の例です。どのようにして描かれたかはこの文書のソースを見れば分かります。

次の例は矢印です。

```
\special{pdf: content
  0.4 0.4 0.6 RG 0.6 0.6 0.8 rg
  2 w
  0 10 m 20 10 l 20 0 l 40 15 l
  20 30 l 20 20 l 0 20 l
  b
}
```



^{*1} 最後の ‘>’ は ASCIIHexDecode フィルタにデータの終りを示すマークです。

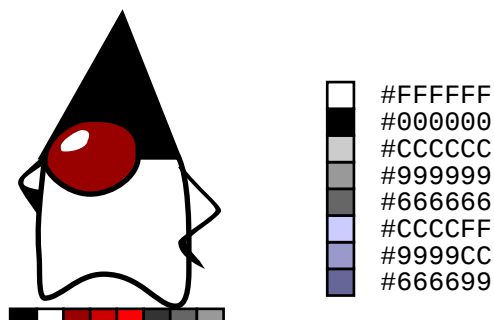


図 2: インデックスカラーを使った例。

2 カラー

PDF はグレースケール、RGB、CMYK など通常の入出力デバイス (装置) で利用されているもののほかにもいくつかの色空間をサポートしています。PDF では文書の体裁だけではなく、『色』もさまざまな環境下でできる限り正確に再現できるように配慮されています。

PDF 1.3 では以下の色空間が利用可能です。

DEVICE グレースケールと RGB、CMYK で、**DeviceGray**, **DeviceRGB**, **DeviceCMYK** があります。これらはデバイス依存だといわれています。ことなる環境下でも同じように再現されるとは限りません。

CIE-BASED **CalGray**, **CalRGB**, **Lab**, **ICCBased** があります。CIE 1976 $L^*a^*b^*$ 色空間や ICC プロファイルに基づく色空間などでデバイス非依存となるように配慮されたものです。色彩工学の研究を通して国際照明委員会 CIE が定めた規格に基づいた色空間ということだそうです。

SPECIAL 色空間といってよいのか分かりませんが、**Indexed**, **Pattern**, **Separation**, **DeviceN** があります。インデックスカラーやパターンなどが含まれます。

これらのうち $\text{T}_{\text{E}}\text{X}$ で処理される文字列に対して適用可能なのは **DEVICE** に分類されている色空間だけです。そのほかにも多少複雑にはなりますが、**special** の中でなら使うことはできます。



図 3: CIE Lab では L^* , a^* , b^* の三つの要素の値を指定することで色が決まる。この例では L^* は固定されており、右方向に向かって a^* が増加し、上方向に向かって b^* が増加するようになっている。

例えばインデックスカラーを使ってみましょう (図 2)。これはカラーパレットを想像されると良いでしょう。この例では **DeviceRGB** 色空間上で選んだいくつかの色に番号がつけられ、この番号を通して色が指定されます。

次の例は少し複雑です。図 3 はパターン (*Shading Pattern*) の例です。色空間には **Lab** が使用されています。使われているパターンそのものは非常に単純なものです。

色空間の切替えは **CS** および **cs** オペレータで行ないます。**sc**, **SC**, **scn**, **SCN** で色の切替えを行ないます。ただし、**DEVICE** に分類されるものでは **G**, **g**, **RG**, **rg**, **k**, **K** という簡略化されたオペレータが用意されていて通常はこれを使います。

例えばこのページのリソース辞書で図 2 の右側にあるインデックスカラーに **Cs1** という名前が関連づけられているとしましょう。このとき、

```
\special{pdf: content
/Cs1 CS /Cs1 cs 0 SC 1 sc
0 0 10 10 re b
0.4 0.4 0.6 RG 0.6 0.6 0.8 rg
10 0 10 10 re b
}
```

とすると  が描かれます。